

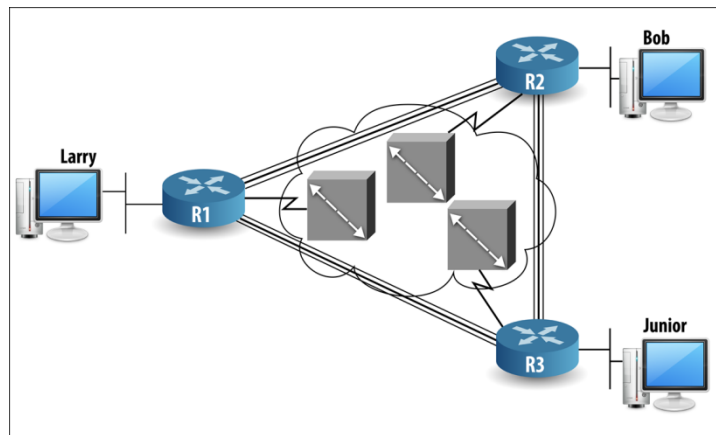


Figure 68.3: Frame Relay Full Mesh

Topic 69: Point-to-Point WANs & Frame Relay

In this topic, we compare point-to-point WANs and Frame Relay.

Frame Relay is considered to be faster, cheaper, and better than using leased lines for a WAN. Point-to-point WANs tend to require more hardware than does the equivalent network built with Frame Relay. Let's consider what the telco does to create three leased circuits. A router requires two serial interfaces and two separate channel service/data service units (CSU/DSU). The telco has to install two cables between the local CO and the office building in Atlanta where R1 resides.

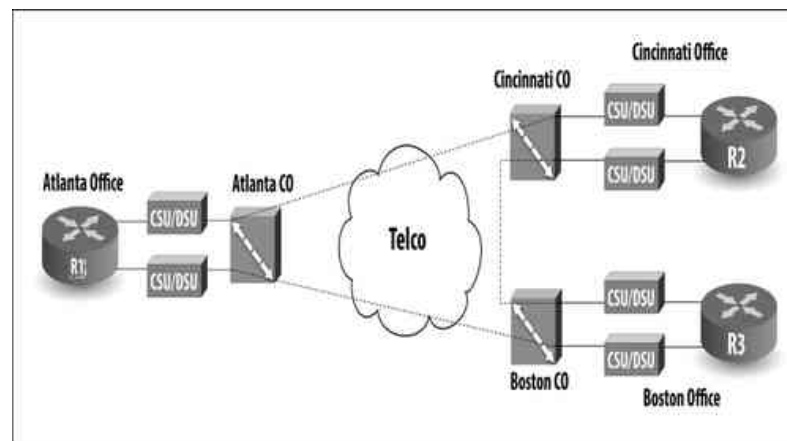


Figure 69.1: Three Leased Lines to Connect Three Routers

Next, we show the equivalent Frame Relay network for the three sites. One cable from each router is needed to the telco, one serial interface, and one CSU/DSU at each router.

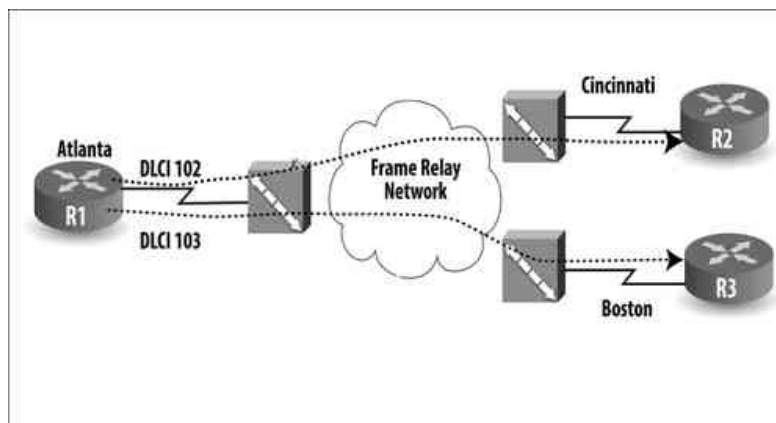


Figure 69.2: Typical Frame Relay Network with Three Sites

What happens when the company grows to 10 sites or 100 sites and employs point-to-point WANs? For each point-to-point line, R1 will need a separate physical serial interface and a separate CSU/DSU. This requires a lot of extra money for router and CSU/DSU hardware. The telco has to run many more cables and more cost.

Topic 70: Routing over PVCs

This topic explains routing over permanent virtual circuits (PVCs).

Assume an internetwork that uses three routers at three sites. Each has an access link to a local CO, and there is no PVC between R2 and R3, making the network a partial mesh of PVCs. The Frame Relay provider (the telco) tells the customer to configure R1 so that it uses data-link connection identifier (DLCI) 102 to send frames to R2, and DLCI 103 to send frames to R3.

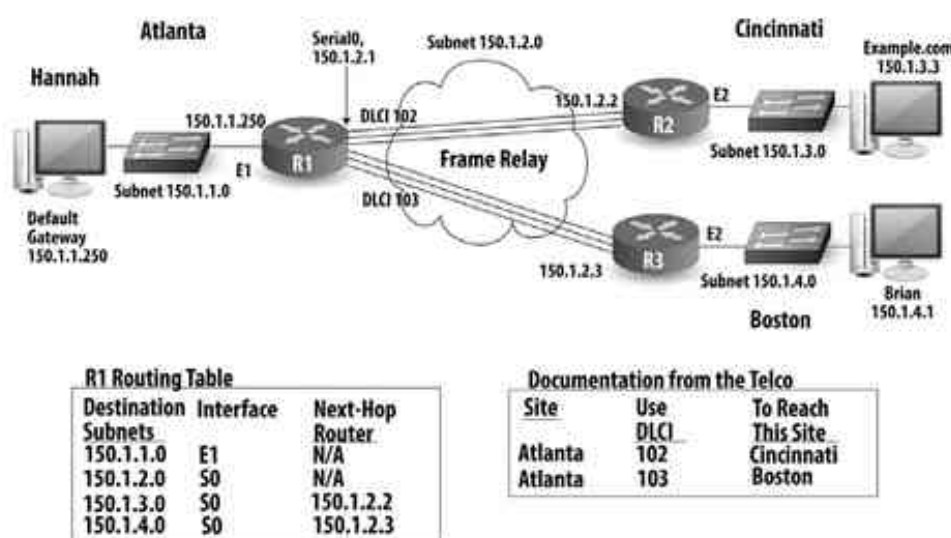


Figure 70.1: Routing over PVCs

Now assume Hannah opens a browser to connect to the `www.example.com`. After using DNS to discover that `www.example.com`'s IP address is `150.1.3.3`, Hannah sends a packet to `150.1.3.3`. Because `150.1.3.3` is in a different subnet, Hannah's PC sends the packet to her default gateway namely, R1. When R1 receives the Ethernet frame, it checks to see if errors occurred. If no error, then R1 performs Frame Relay encapsulation. Frame Relay header holds the 10-bit-long DLCI field that identifies the PVC. The trailer has a frame check sequence (FCS).

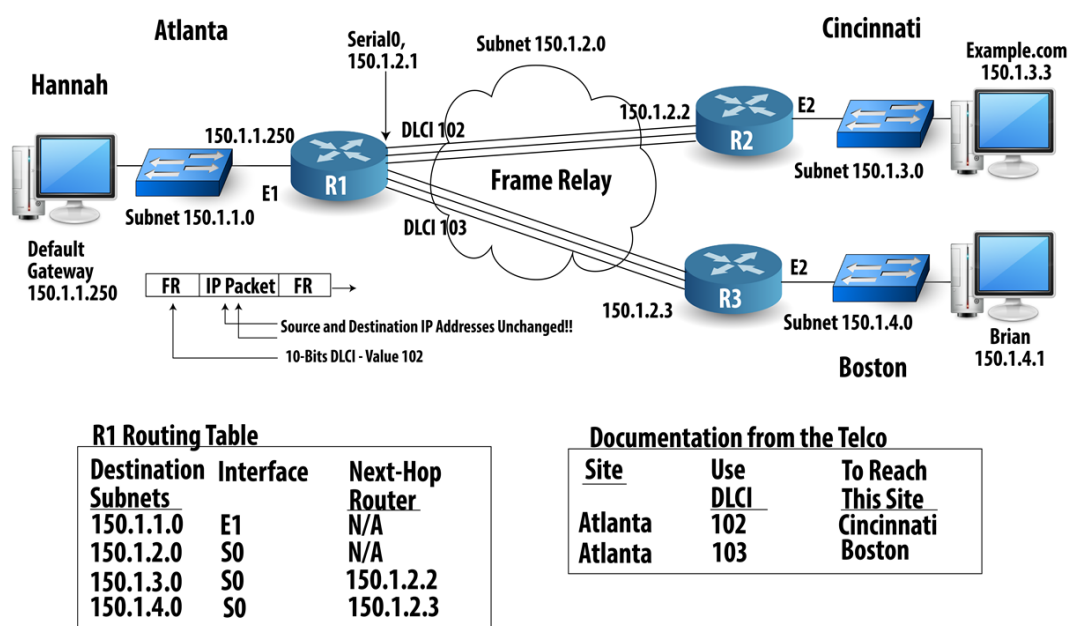


Figure 70.2: Forwarding an IP Packet over a Frame Relay PVC

When R1 looks at his routing table for subnet `150.1.3.0`, it lists next-hop router of `150.1.2.2`, which is R2's IP address, and outgoing interface `S0`. It does not list which DLCI to use to get to `150.1.2.2`. As soon as the PVC starts working, R2 announces its IP address to R1, using the VC between the two routers. R1 also announces its IP address to R2, using that same VC. In this way, both routers learn the other router's IP address that is used on that VC. The message used to announce the IP address and DLCI is called an Inverse ARP message.

Topic 71: The Internet: A Large IP Network

This topic explains why connecting to the Internet provides a path for IP packets to be sent almost anywhere.

A company cannot run a cable between two far-away buildings for legal and practical reasons. A telco already has the ability to let you connect to its network physically,

and then deliver the data over its network. With WAN links, the telco allows two sites to communicate. With Frame Relay, many sites can communicate, but they all must use the same Frame Relay service from the same telco. The company can connect to the Internet with a WAN link or Frame Relay. With that one physical connection, that company gains access to a huge number of people who also have access to the Internet. Next, we show an example depicting reaching the world through one connection to the Internet.

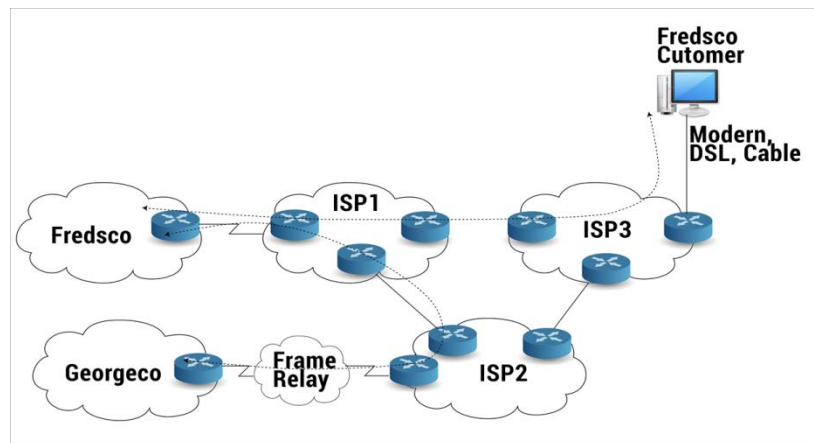


Figure 71.1: Reaching the World through One Connection to the Internet

An Internet service provider (ISP) creates an IP network over which IP packets can be forwarded, similar to the way a telco creates a network over which bits (for WAN links) and Frame Relay frames (for Frame Relay) can be forwarded. Some ISPs are also telcos. The Internet consists of three major components:

- The IP networks that ISPs create.
- The enterprise IP networks that attach to one or more ISPs.
- The individuals who connect their computers to ISPs.

WAN Type	Generic Name of Provider	Physical Connectivity	Description of Service
WAN link (leased line)	Telco	4-wire cable from customer to telco CO at each site	Telco forwards bits between two sites
Frame Relay	Frame Relay service provider	4-wire cable from customer to telco CO at each site	Provider forwards frames, based on DLCI
Internet	Internet service provider	Either WAN link or Frame Relay bet. customer and ISP	ISP routes IP packets to any IP host on the Internet

Table 71.1: Comparisons of WAN Links, Frame Relay and the Internet

Topic 72: Using a Phone Line for Data

In this topic, we describe how a modem can be used to transmit data from home.

A telephone includes a microphone, which converts sound waves into an analog electrical signal. It is carried by a local loop (phone line between a house and a telco CO). The telco can send the electrical signal between one phone and another. On the receiving side, the phone converts the electrical signal back to sound waves using a speaker. The characteristics of the analog signal are:

- Frequency: the number of times a signal repeats itself, from peak to peak, in one second.
- Amplitude: how strong the signal is.

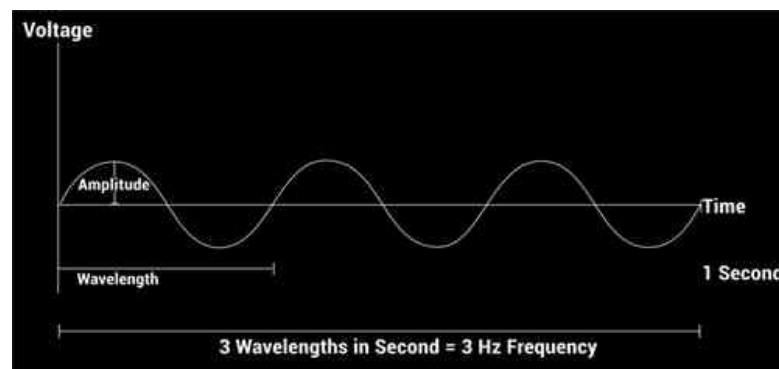


Figure 72.1: Analog Electrical Signal: Frequency and Amplitude

The sounds that your voice makes happen to be a continuously changing sound wave, so analog electrical signals that continuously change work well for voice. Modems allow two computers to send and receive a serial stream of bits over an analog phone circuit, with no physical changes required on the local loop between a residence and the telco CO. As the telephone switch in the CO expects to send and receive analog voice signals over the local loop, modems simply send an analog signal to the PSTN and expect to receive an analog signal from the PSTN. Instead of voice, a modem converts a string of binary digits on a computer into a representative analog electrical signal.

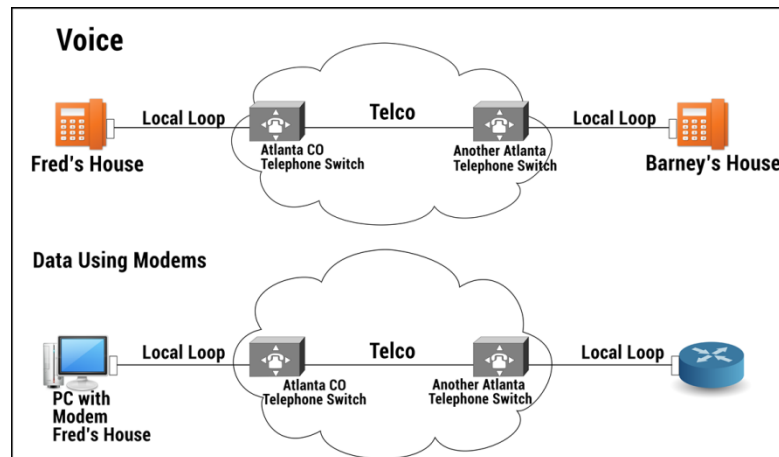


Figure 72.2: Comparing a Phone to a Modem

Modems encode a binary 0 or 1 onto the analog signal by varying frequency or amplitude. Changing these characteristics of the analog signal is referred to as modulation. Basic Operation of Modems over PSTN are shown next.

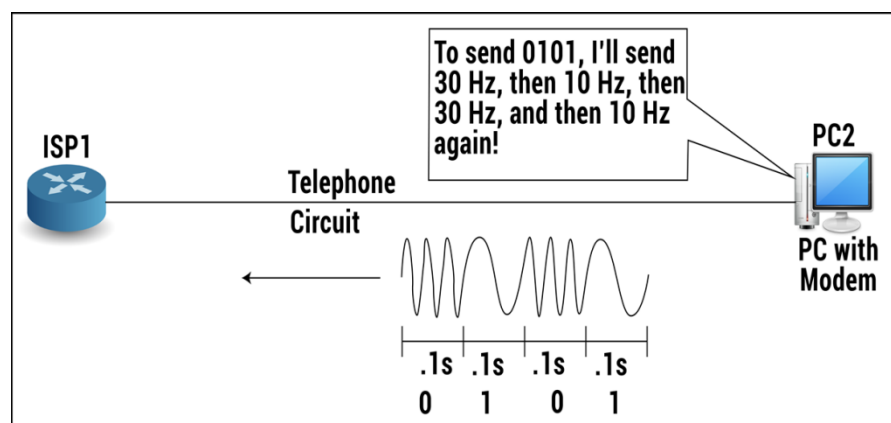


Figure 72.3: Basic Operation of Modems over PSTN

In previous example, frequency changes every .1 seconds, PC can send 10 bits/sec. If a modems sends at 9600 bps, the receiving modem needs to sample the incoming analog signal every $1/9600$ of a second. The details of modem relate to OSI Layer 1, the physical layer.

Topic 73: Digital Subscriber Line (DSL)

In this topic, we describe the usage of DSL for Internet.

Modems use the phone line to make what the telco thinks is a voice call, so you cannot make another call while you are surfing the Internet. Another big negative is the speed. Digital subscriber line (DSL) is an alternative technology for sending and receiving data to and from an ISP, using the same old phone line, but running at much faster speeds. Also, you can make phones calls and surf (send IP packets) at

the same time. But it is a little more expensive. DSL allows the same old analog voice signal to be sent over the line by a phone. At the same time, DSL allows a separate digital signal to go over the same phone line. The telephone generates analog signals ranging from 0 to 4000 Hz. Most speech ranges from 300 to 3300 Hz. The digital DSL signal uses frequencies higher than 4000 Hz – do not interfere with speech/hearing. DSL uses the same local loop wiring that is already installed between the CO and your house, but now the CO connects the local loop wiring to a device called a DSL access multiplexer (DSLAM). The DSLAM splits out the digital signal and analog signal from the local loop. The DSLAM gives the analog voice signal to a telephone switch while forwards the data traffic to a router.

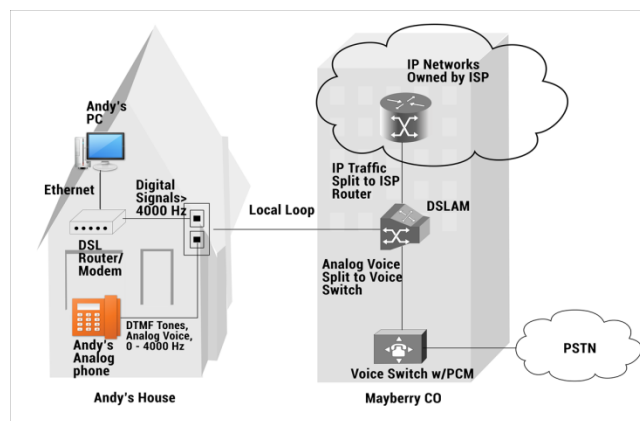


Figure 73.1: DSL Connection from the Home

Some DSL variants allow the local loop to be much longer than some other DSL variants, whereas others only allow for a shorter local loop. For a shorter local loop, transmission rates are much higher – design tradeoff. The speed at which data goes from the Internet to the home is faster than the other direction a feature called asymmetric transmission rates. More bandwidth is needed for downloading.

Figures and Material used for this topic have been adapted from Kurose and Ross's book with the title *"Computer Networking: A Top-down Approach"*, 6th Edition, 2013.

Topic 74: Sending Data without a Phone Line

In this topic, we describe how TV cable networks can be used for the Internet.

Cable Internet access makes use of the cable television company's existing cable television infrastructure. A residence obtains cable Internet access from the same

company that provides its cable television. A cable head end broad casts television channels.

Cable Network Architecture: Overview

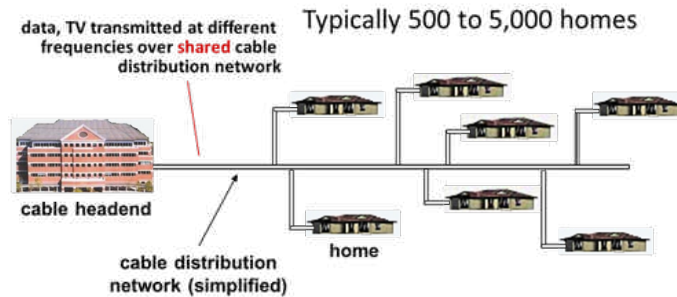


Figure 74.1: A hybrid fiber-coaxial access network

Fiber optics connects the cable head end to neighborhood-level junctions, from which traditional coaxial cable is then used to reach individual houses. This is called Hybrid fiber coax (HFC) and is shown next in detail.

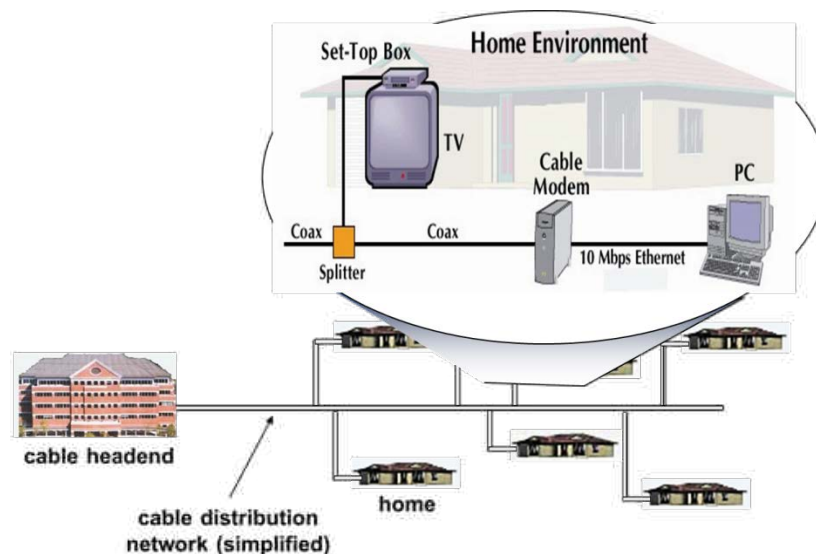


Figure 74.2: A hybrid fiber-coaxial access network

Frequency division multiplexing: different channels transmitted in different frequency bands.

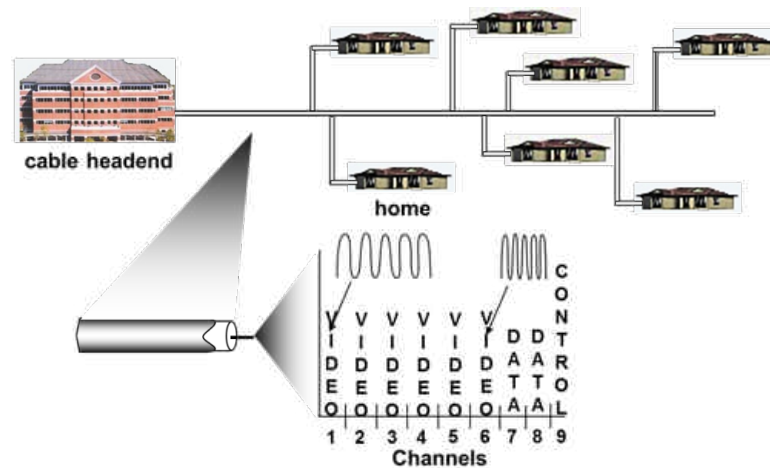


Figure 74.3: A typical home network

Topic 75: Introduction to AAA Security Model

In this topic, we list and describe the three components of the AAA security model.

AAA stands for authentication, authorization, and accounting. The AAA model refers to three areas of security. All three relate to issues regarding an individual device or user. Before you can use most servers, you need to have the right to use that server. A username and a password are required. **Authentication** is a five-step process.

- 1. Fred requests a web page.
- 2. The web server replies, asking for a username, password.
- 3. Fred types in and sends his username, password to server.
- 4. The web server checks its list of usernames and verifies that the password is correct.
- 5. If the username and password are correct, the web page returns the contents of the web page.

The working of basic authentication process is shown next:

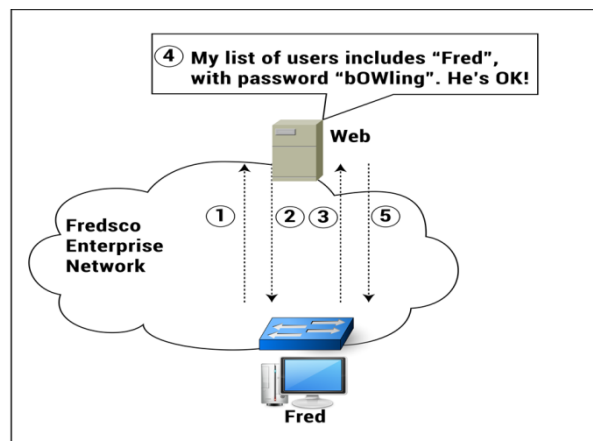


Figure 75.1: Basic Authentication Using a Username and Password

Username, password can be listed on a server called authentication server. Application servers query authentication server. Kerberos protocol runs between application and authentication servers. This is shown next.

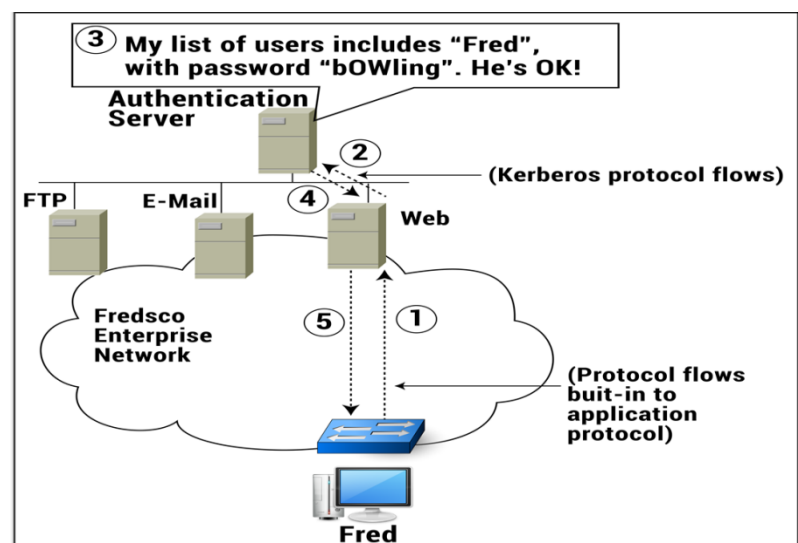


Figure 75.2: Basic Authentication Using a Username and Password

The second "A" in AAA is for **Authorization**. It refers to the process of figuring out what a particular user is allowed to do.

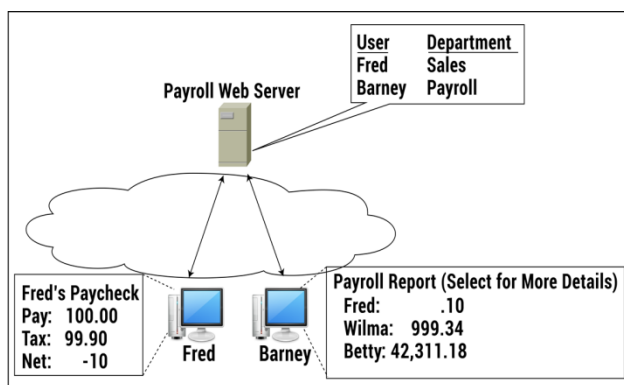


Figure 75.3: Fred Can't See Others' Payroll Information

The last "A" in AAA is **accounting**. The same servers that perform authentication and authorization services can keep a record of each request to authenticate or authorize a user. Accounting allows to record and report when users type the wrong password. Several consecutive attempts to connect to a server with an invalid password each time signals someone is trying to guess Fred's password. Accounting features can record individual attempts, generate reports, and notify personnel if too many invalid attempts happen in a short timeframe.

Topic 76: Password Authentication Protocol

In this topic, we describe the working of password authentication protocol (PAP).

Unlike those people who are attached to a LAN in a physically secure office building, people who use modems or DSL to connect to the Internet from home have to go through an additional authentication step. ISPs authenticate users before they can even use the network. This helps ISPs to determine whether a customer has paid his bill or not. Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP) are used by ISPs. Both are part of Point-to-Point Protocol (PPP) — a data link protocol. Let's assume that Fred dials an ISP with a modem, PAP is used between himself and the ISP router. The following sequence is used:

- 1-Fred uses PAP to send a username, password.
- 2-The router sends a request to an authentication server using RADIUS protocol messages.
- 3-The authentication server checks the username, password.
- 4-The authentication server confirms that Fred is authentic using RADIUS.
- 5-The router uses PAP to confirm that Fred is allowed to use the Internet.

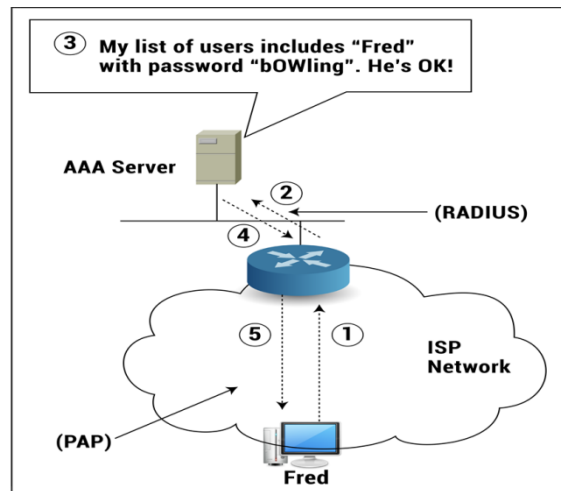


Figure 76.1: Basics of PAP

TACACS+ is a popular proprietary protocol that Cisco developed before RADIUS existed and can be used in place of RADIUS. Also, CHAP can be used instead of PAP. PAP sends the username and password in clear-text. Anyone with the right tools can actually read your username and password. To protect against password theft, CHAP does not send the password as clear-text. Process of CHAP starts as soon as a modem calls a phone number at the ISP or DSL modem is powered on. CHAP uses six steps to authenticate user.

- 1- Router generates a random number and sends it in a CHAP message to the PC.
- 2-PC runs a math function with the random number and the password typed by the user as input.
- 3-The PC sends the results of the function called a message digest back to the router.
- 4-The router sends the username, the random number, and the message digest to the AAA server.
- 5a-The AAA server uses the same math that the client used in Step 2, with the same random number, plus the password associated with that username in the AAA user database.
- 5b- The result is another message digest.
- 5c-If the message digest calculated by the AAA server matches the one calculated by the PC, the password that the user typed must be the right one.
6. The AAA server tells the router that the user is authentic; the router tells the PC, and life goes on.

The password was never passed through a network connection during this procedure. This is shown next.

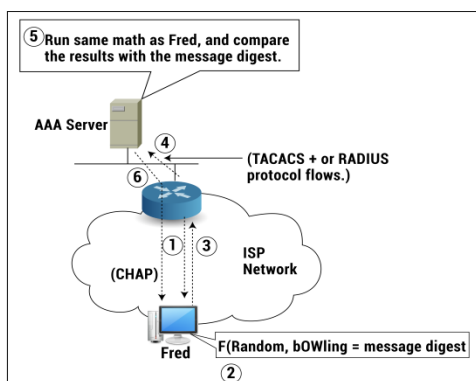


Figure 76.2: Hashing Passwords to Create a Message Digest

Topic 77: Virtual Private Network (VPN)

This topic explains benefits of using a virtual private network (VPN).

Encryption allows a computer to apply a mathematical formula to data, sending the results of the mathematical function over the network. The data just looks like a bunch of random bits. The computer receiving the data can then re-create the original data by decrypting the data. To decrypt the data, you need a secret password called an encryption key. These days, it is common for users to encrypt data before sending it over the Internet. Instead calling it encryption; it is called a virtual private network (VPN). The packets go across the Internet, it is still a physical public network. VPNs create a private network, but they do so logically, or virtually.

Let's assume Barney works for a company. The enterprise network at Barney's company is a private network, with all the components inside privately controlled office space. The Internet is public. For Barney to use the VPN, he must encrypt the packet as he creates it. To do this, Barney needs to have VPN software installed on his computer. He also needs to know the encryption key to use. A VPN device inside the corporate network is called a VPN concentrator, which decrypts packets received from Barney and others. This is shown next.

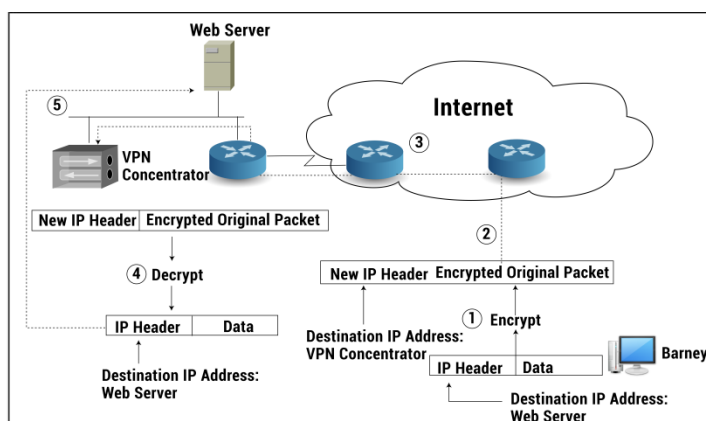


Figure 77.1: Encrypting IP Packets for a VPN

Topic 78: Enterprise Network and the Internet

In this topic, we list the traffic that should and shouldn't be allowed between an enterprise network and the Internet. When you connect an enterprise network to the Internet, one of the first things you must decide about is what you want to allow to pass to and from the Internet, and what you don't. Assume an enterprise network 'Internal IP Network'. There is an internal web server that has stuff only appropriate for employees who work for Fredsco. The external web server is meant for external users, but internal clients will also want to browse that web server.

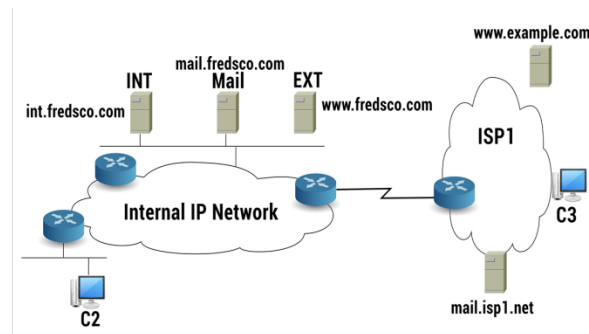


Figure 78.1: An Enterprise Network Connecting to the Internet

To secure Fredsco's network, two things have to be kept in mind: Between which two hosts do packets need to flow? Which host begins that communication? To secure Fredsco's network, two things have to be kept in mind: Between which two hosts do packets need to flow? Which host begins that communication? Typical types of traffic allowed are shown next.

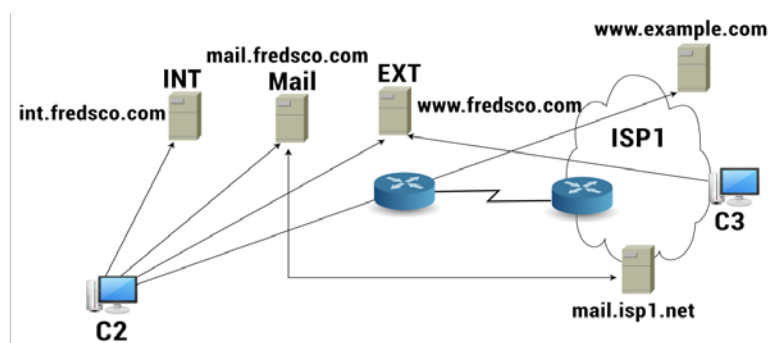


Figure 78.2: Typical Types of Traffic Between an Enterprise and the Internet

The bad guys on the Internet who are trying to get into Fredsco's network should be prevented. Traffic that is typically not allowed is shown next.

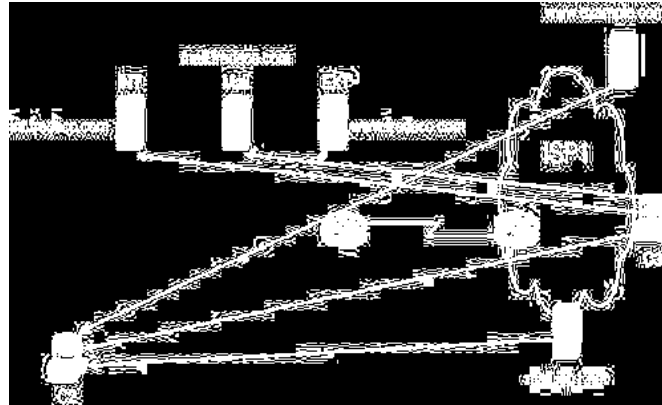


Figure 78.3: Traffic That's Typically Not Allowed

Topic 79: Firewalls, Demilitarized Zone, IDS

In this topic, we describe the use of firewalls, demilitarized zone (DMZ) and intrusion detection systems (IDS).

A network engineer configures the firewall with a set of rules that tells it what's legal and what isn't. Then the firewall allows some packets to pass through it and discards others to enforce the rules. The firewall needs to be in the path that is used for forwarding packets to and from the Internet. Cisco's firewall product is called a PIX firewall.

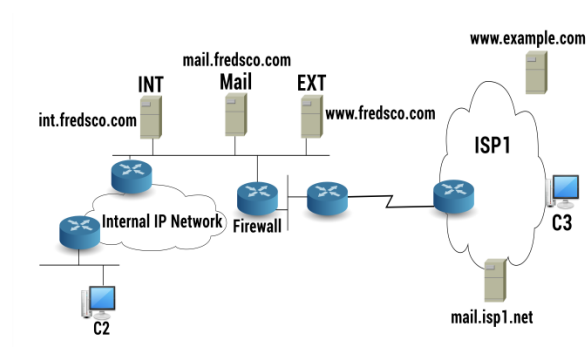


Figure 79.1: Putting Up a Wall Between the Dangerous Folks and Your Network

As a firewall watches the traffic entering the network, it knows the nature of the traffic that is allowed to flow through it. Also, it recognizes when a host is initiating a new flow. A host who is initiating a new flow can be recognized by looking at: The first TCP segment used to create a TCP connection SYN flag bit =1, and source IP address of the packet.



Figure 79.2: TCP Connections and Well-Known Ports

The scenario depicting the allowance of the TCP Connection is shown next.

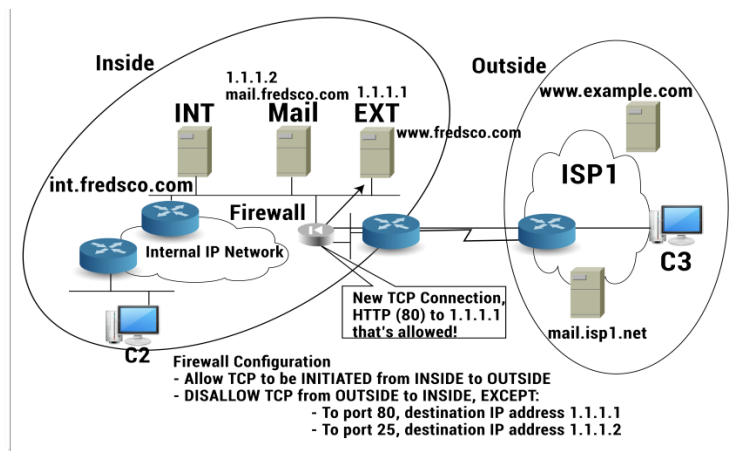


Figure 79.3: Allowing the TCP Connection from Figure 79.2

The firewall uses a similar logic to stop packets that should not be allowed. This is shown next.

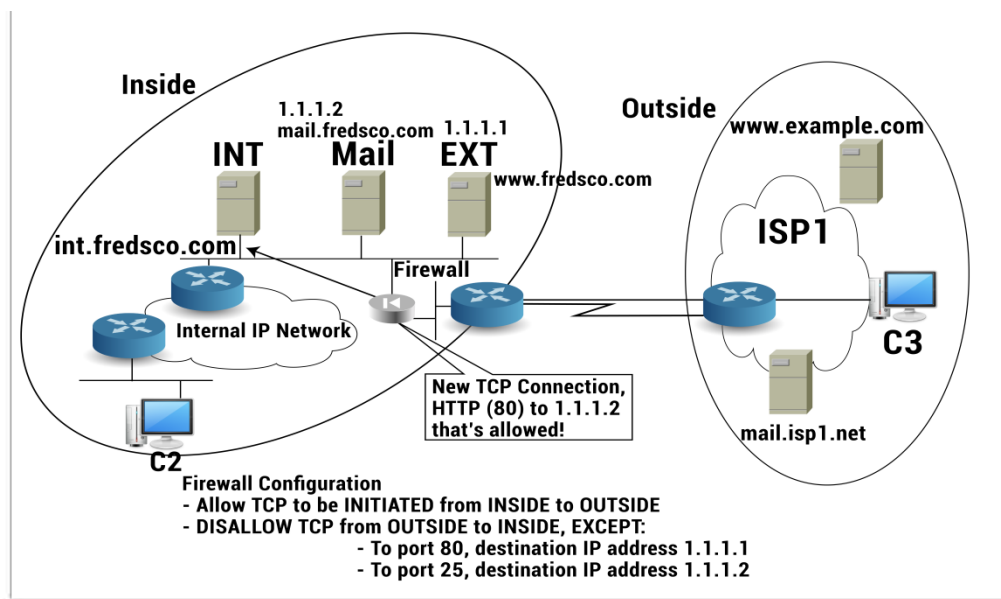


Figure 79.4: Disallowing a TCP Connection to an Inside Web Server

Firewalls are like routers. They forward packets based on destination IP address. They have at least two physical interfaces. They can have more than two interfaces. Outside interface connects to Internet. Inside interface connects to internal network.

A third interface connected to a LAN called a demilitarized zone (DMZ), somewhere bet inside & outside interfaces. With a DMZ, Internet-accessible servers can be placed on a different LAN. A stronger firewall rule: No TCP connections can be initiated from outside to the inside. The only flows allowed are for servers in the DMZ.

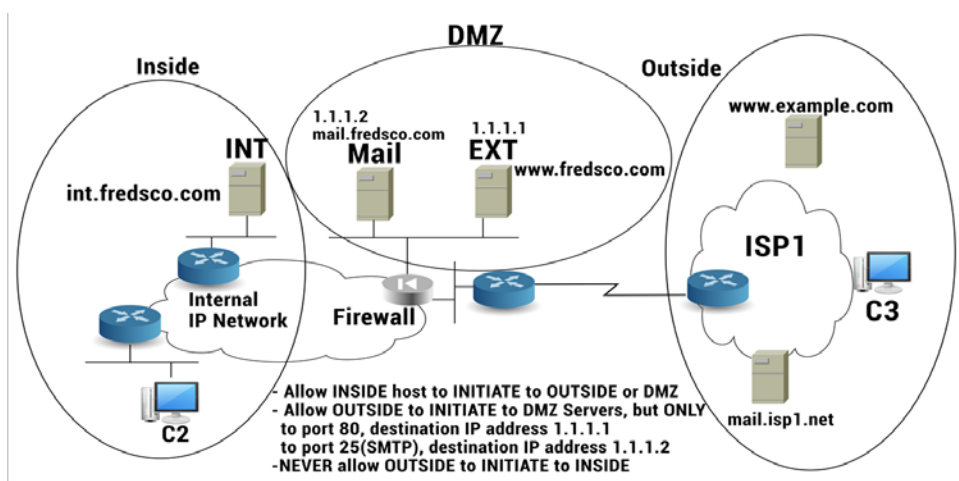


Figure 79.5: A Safe but Potentially Risky Place: The DMZ

Intrusion detection systems (IDSs): watch packets that a firewall allows through. They look for things in the packets to determine if someone is cheating firewall and do bad things to servers in network. Some IDS devices sit in the network, watching packets that pass over a LAN and are called network-based IDSs. Those IDS softwares that sit on the servers are called host-based IDSs. A network-based IDS is shown next.

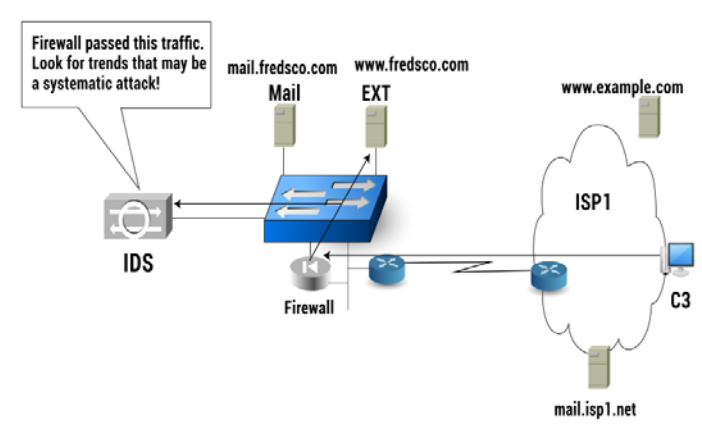


Figure 79.6: Watching for Patterns with a Network-Based IDS

Handouts for Part 2 (Topics 80 to 120) CS206: **Introduction to Network Design and Analysis**

Figures and Material used for this topic have been adapted from “Wireshark User’s Guide”, U. Lamping, R. Sharpe, and E. Warnicke, 2014, https://www.wireshark.org/docs/wsug_html/.

Topic 80: Introduction to Wireshark

This topic provides introduction to Wireshark.

Wireshark is a network packet analyzer which captures network packets and displays that packet data as detailed as possible. Wireshark is a free open source software program available at www.wireshark.org. G. Combs is the original creator of Ethereal (Wireshark's development name prior to May 2006). As an open source project, it's source code is open.

Intended Purposes: When run on a host connected to a wired or wireless network, Wireshark captures and decodes the network frames. People use it to learn network protocol internals. Network administrators use it to troubleshoot network problems. Network security engineers use it to examine security problems. Developers use it to debug protocol implementations.

Wireshark’s Features

- 1- Available for UNIX and Windows.
- 2- Capture live packet data.
- 3- Display packets with very detailed protocol information.
- 4- Filter packets on many criteria.
- 5- Save packet data captured.

Wireshark’s Installation

Get Wireshark installer from: www.wireshark.org/download.html and execute it.



Figure 80.1: Wireshark's Installation wizard

Installation Components

- **Wireshark** - The network protocol analyzer.
- **TShark** - A command-line network protocol analyzer.
- **Plugins & Extensions** - Extras for the Wireshark and TShark dissection engines.
- **Tools** - Additional command line tools to work with capture files
- **User's Guide**



Figure 80.2: choose components

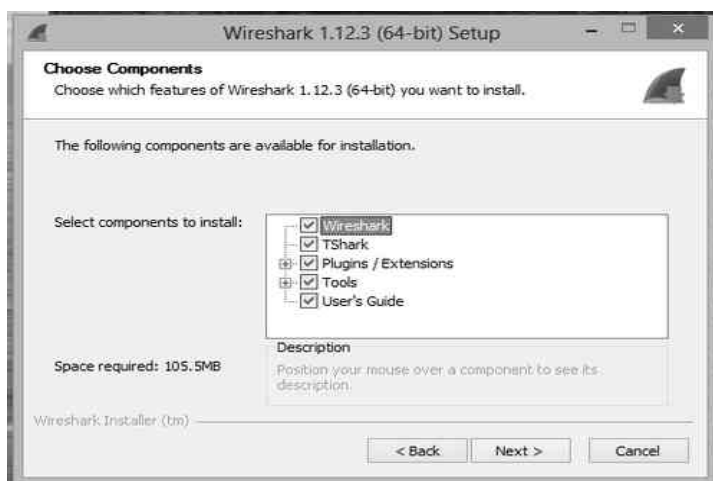


Figure 80.3: choose components

Installing WinPcap: With WinPcap installed you would be able to capture live network traffic.



Figure 80.4: Installing WinPcap



Figure 80.5: Installing WinPcap

The main window is shown next.

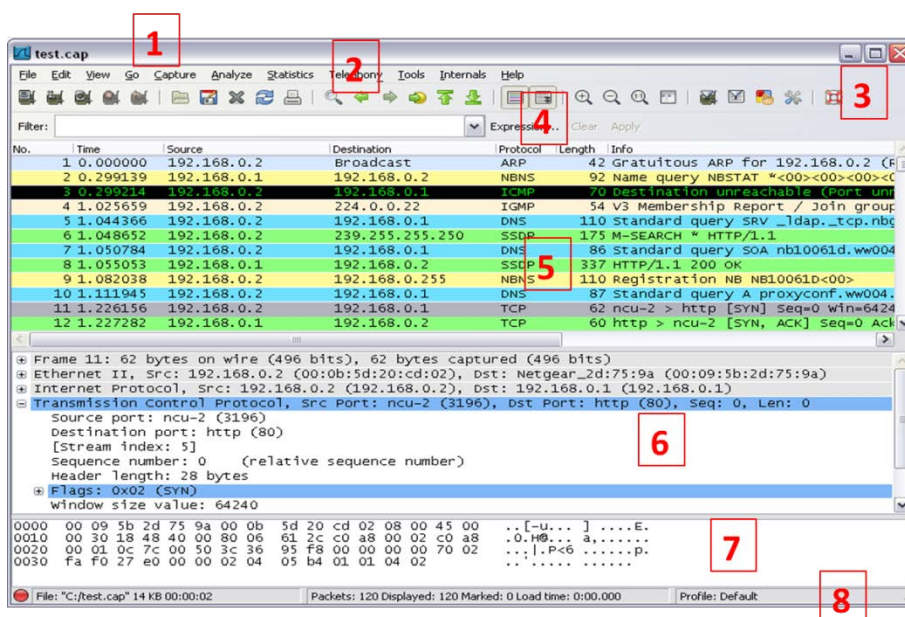


Figure 80.6: The Main window

Area 1 shows the Title bar — trace file name. Area 2 represents Main menu — standard menu. Area 3 is the Main toolbar — provides quick access to frequently used items from menu. Area 4 depicts Display filter area — reduce amount of traffic you see. Area 5 shows Packet List pane — summary of frames. Area 6 is the Packet Details pane — dissected frames. 7. Packet Bytes pane — hex/ASCII details. Area 8 represents Status Bar — access to the Expert, annotations, packet counts, and profiles.

Figures and Material used for Part2 (from Topics 81 to 172) have been adapted from “Troubleshooting with Wireshark: Locate the Source of Performance Problems”, 2014, by L. Chappell.

Topic 81: Five Common Network Problems

This topic discusses 5 common Network problems.

TCP Connection Refused by Server

A service is not running on a server or perhaps a firewall is preventing the connection. TCP connection refusals may also be due to a closed port. Possible Symptoms: Client's SYN followed by Reset (RST)/ACK. Client SYN followed by an ICMP Destination Unreachable/Port Unreachable response from a host-based firewall.

Connection Blocked by a Host-Based or Network Firewall

Ideally, hosts would not even attempt to communicate with firewalled resources. Such an attempt could be due to a misconfiguration, malware, malicious user, or other issue. Possible Symptoms: No response to a SYN packet, RST/ACK response to a SYN packet.

Slow Application at Server

The good news is that the server did not refuse to provide a desired service. The bad news is that the server is slow. This may be due to: 1- lack of processing power at the server, 2-a poorly behaving application, or 3-even, in a multi-tiered architecture, a slow upstream server that actually provides the data. Possible Symptoms: TCP-based app: Large delay between the server ACK to a client request and the response data. UDP-based app: Large delay between a request and the response data.

Slow Load of Remote Content

Many networks are designed in a multi-tiered fashion. For example, consider a client that sends a request to Server 1. That server may need to obtain information from Servers 2 through 9 before answering the client. We need to capture this multi-tiered traffic to determine which server is actually responding slowly. Possible Symptoms: TCP-based app: Large delay between the server ACK to a client request and the response data. UDP-based app: Large delay between a request and the response data.

Topic 82: Next 5 Common Network Problems

This topic describes next 5 common Network problems.

Server Application Fault

The server is up and running, but not responding to requests. The server responds to a SYN with a SYN/ACK, but when the client makes a request, no response is received. When the TCP Retransmission Time Out is reached, client retransmits request. If no ACK is received, it continues to retransmit request using an exponential backoff time until it finally gives up and sends a TCP Reset. Possible Symptoms: Retransmissions.

Content Redirection

Upon arriving at a store to buy something, we might be told that the store does not have the item in the stock; please go to another store to get it. If an application supports redirection (such as HTTP) and the target knows where the information actually resides, it may send you a redirection response. Possible Symptoms: In HTTP communications, response codes 300-399. A sudden “name resolution process” and traffic to another host.

TCP Receive Buffer Full

When advertised TCP receive buffer value drops to zero, data transfer will stop. A low Window Size value can also cause data flow to stop. The only recovery is a Window Update. Possible Symptoms: Unusually high TCP delay before a Window Update and resumption of data flow.

TCP Send Buffer Full:

A limited send buffer space can negatively impact network performance even if network can handle a high data transfer rate and the receiver has plenty of buffer space. Possible Symptoms: High TCP delay for no apparent reason.

Altered TCP Attributes along a Path

Routers and other interconnecting devices can bring havoc on a network if they change the attributes of a TCP connection as they forward the traffic. Possible Symptoms: Low TCP calculated window size and significant delays before Window Update packets.

Topic 83: Next 6 Common Network Problems

In this topic, we provide description of next 6 Common Network Problems.

Mismatched TCP Parameters across a Proxy Device

Connection parameters on one side of a proxy device do not match the connection parameters offered in connection on other side of proxy device. Possible Symptoms: Delays in data forwarded through proxy (proxy queuing).

Routing Loops

Routing loops occur when a packet is routed back onto a network over and over again. Possible Symptoms: Identical packets listed but no TCP Retransmission indications (they are not identical packets—their TTL value decrements).

Weak Signal (WLAN)

If a wireless signal degrades substantially, it may not be interpreted properly when captured. Possible Symptoms: Low Signal Strength value.

Asymmetric Routing

This is a situation when traffic flowing from Host A to Host B flows along a different path than the traffic flowing from Host B to Host A. This can be a problem, if there are devices that must see every packet in order to function. For example, network path must be symmetrical from a client to a proxy host. An IDS box must see every packet. Possible Symptoms: ACKed segment that was not captured.

Packet Loss

Typically, this occurs at an interconnecting device such as a switch, a router, a NAT device, or a network firewall. If the number of packets dropped is small and the recovery process is quick, the packet loss may go unnoticed. If many sequential packets are lost, however, users will likely feel the impact and complain. Possible Symptoms: Fast Retransmission. Duplicate ACKs.

High Path Latency

A single low speed (high delay) link along a path or the delay between geographically disbursed peers can inject a level of path latency that affects performance. Possible Symptoms: Large delays between the outbound SYN and the inbound SYN/ACK of a TCP handshake.

Topic 84: Next 4 Common Network Problems

This topic discusses Next 4 Common Network Problems.

Lousy Routing Path:

When construction occurs in a major city, traffic is a nightmare as drivers are rerouted along less efficient and less direct routes. This is also true of network traffic. If a target is 10 blocks away and yet for some reason the packets must travel through 17 routers to get there, performance may be unacceptable. **Possible Symptoms:** Large delays between the outbound SYN and the inbound SYN/ACK of a TCP handshake.

Bandwidth Throttling

Transmitting data along a bandwidth throttling link is like driving a car during rush hour. You move along bumper-to-bumper at speeds below your lowest speedometer indications.

Delayed ACK

It is defined in RFC 1122 as a method to "increase efficiency in both the Internet and the hosts by sending fewer than one ACK segment per data segment received." This delay can cause issues for the host that is sending data if it times out waiting for an ACK or it cannot send another data packet until an ACK is received. Possible Symptoms: 200ms delay before ACK packets.

Queued Packets (Overloaded Router)

TCP peers and UDP-based applications may detect sudden queue delays and think packets have been lost. This may be caused by an overloaded router or perhaps prioritization at a router (e.g., video streaming first, email traffic last). Possible Symptoms: Decrease in throughput followed by equal increase in throughput.

Topic 85: Next 5 Common Network Problems

This topic studies Next 5 Common Network Problems.

Route Redirections

When a host sends packets to one local router when another local router exists with a preferred path, a route redirection may take place. The receiving router may respond with an ICMP packet that includes IP address of the recommended router. Rare - we see only one router on the local network. Possible Symptoms: ICMP Type 5 packets with either Code 0 (Redirect for Host) or Code 1 (Redirect for Network) indicate route redirection is taking place.

Broadcast or Multicast Storms

Broadcasts are typically not forwarded so you should be capturing on and dealing with hosts on a single network. Multicasts can be forwarded through an internetwork therefore they can cause a greater problem if something goes wrong and they storm the network. Possible Symptoms: High rate of packets addressed to the all nets broadcast (255.255.255.255). High rate of packets addressed to a multicast address (224-239.x.x.x).

Switch Loop

Older switches that have not been restarted or rechecked every once in a while may begin to malfunction. If there is a loop—network will become overwhelmed with looped packets. Possible Symptoms: High rate of identical packets.

Virus/Malware on Network Hosts

When a compromised host begins performing port scans on other network hosts or it begins broadcasting discovery packets, the overhead may be felt. Possible Symptoms: Local broadcasts. Unusual internal targets (such as a local client trying to connect to the Accounting server).

Network Name Resolution

This is imperative to connecting to a target. A user may receive an application error such as "Server not found" as in the case of a DNS Name Error when browsing. Possible Symptoms: No response to a name query. An error response to a name query (such as a DNS Name Error or Server Error).

Topic 86: Next 8 Common Network Problems

This topic studies Next 8 Common Network Problems.

Network Address Resolution

Incorrect subnet addressing can cause a client to send packets that are destined to a local device to a router. Possible Symptoms: ARP Requests are sent for remote targets. Traffic destined to local hosts are sent to the router.

Hardware Address Resolution

This allows obtaining MAC address of a local target or local router. Network address resolution problems may cause a host to think local devices are remote (or vice versa). Possible Symptoms: No response to an ARP query for a local host. No response to an ARP query for a local router.

No Support for Selective ACK

When packet loss is detected by a receiving TCP host that supports SACK, that host can acknowledge data received after the missing packet(s). Without SACK, the first packet lost and every subsequent data packet will be retransmitted even when those subsequent packets were received successfully. Possible Symptoms: No SACK Option in the TCP header of a SYN packet. No SACK Option in the TCP header of a SYN/ACK packet when there was a SACK Option in the TCP header of a SYN packet.

No Support for Window Scaling

Window Scaling is used to increase the advertised TCP receive buffer size past the 65,535-byte limit caused by the 2-byte Window Size field. During the TCP handshake, peers indicate that they support Window Scaling and provide a Window Scale Shift Count. Reduces the delays caused by a low window or Zero Window condition. Possible Symptoms: No Window Scaling Option in TCP header of a SYN packet. No Window Scaling Option in TCP header of a SYN/ACK packet when there was a Window Scaling Option in TCP header of a SYN packet.

Client Misconfiguration

A misconfigured client may have the wrong DNS address, the wrong router address, incorrect port numbers defined in a services file, or other problems. Possible Symptoms: The client sends traffic to the wrong target. The client receives service refusals or no answer. Numerous other symptoms may appear depending on the misconfiguration.

Low Packet Size/Low MTU Size

Amount of data that can be carried in a frame is limited by MTU, MSS, or even an inefficient application. Performance pains when sending large files. Possible Symptoms: The MSS value defined in the TCP SYN or SYN/ACK packet is illogically small.

TCP Port Number Reuse

Port numbers can be reused without any problem if previous TCP connection is terminated. Otherwise, the new connection is refused. Possible Symptoms: TCP RST sent in response to a SYN packet to a server port that is known to be open.

Slow Application

Slow applications can be due to poor or bloated coding, internal errors, or even man-made timers defining the application's performance speed. Possible Symptoms: Large delay in the application response time value.

Topic 87: A Four-Part Analysis Methodology

In this topic, we explain an analysis technique for the troubleshooting of networks.

A Network Analyst's Perspective includes Task 1: Define the problem, Task 2: Collect system, application and path information, Task 3: Capture and analyze packet flows, and Task 4: Consider tools.

Define the Problem

A common complaint from network users: "The network is slow". This is a vague description. Need more information from the user. Q1. What type of traffic will you be looking for? A file upload, file download, login, email send/receive or something else? Q2. What server target were you communicating with? Q3. What are the symptoms? Q4. Did you receive any error message? Q5. Is this happening all the time?

Collect System, Application and Path Information

Obtain as much system, network and infrastructure information as possible. Ask the customer about their configuration. Q1. What operating system is running on the client/server? Q2. What application (and version) is running? Q3. Describe the network path that the traffic must traverse.

Capture and Analyze Packet Flows

Capture Location Tips: Capture as close as possible to the complaining host so that you can observe the traffic from that host's perspective, examine the roundtrip time to the target(s) etc. Capture Tool Tips: Prefer using a tap or dedicated capture device rather than spanning a switch port for the capture process. Analysis Process Tips: Know what is "normal". Remove unrelated traffic from view. Focus on traffic related to the complaining user's machine etc.

Consider Other Tools

Once you have hit a Wireshark limitation, you may need to work with another tool. If size of trace files becomes larger than 100 MB, Wireshark becomes too slow.

Topic 88: Using a Troubleshooting Checklist

This topic describes the usage of a Troubleshooting Checklist.

Captured traffic is saved in a trace file. When you open a trace file, use a basic troubleshooting checklist. The order in which you go through the checklist may change depending on the troubleshooting issue.

Verify Trace File Integrity and Basic Communications

Look for ACKed Unseen Segment. Verify traffic from the complaining user's machine is visible. Verify resolution process completion such as DNS/ARP.

Focus on Complaining User's Traffic

Filter out unrelated traffic. Export related traffic to a separate trace file.

Detect and Prioritize Delays

Sort and identify high delta times. Measure path latency (Round Trip Time) using delta times in TCP handshake. This can be achieved by measuring delta from TCP SYN to SYN/ACK. Measure server response time. This can be obtained by measuring from request to response.

Look for Throughput Issues

Click on low throughput points to jump to problem spots in the trace file. Look at traffic characteristics at low throughput points.

Check Miscellaneous Traffic Characteristics

Check packet sizes during file transfer. Check for ICMP messages. Check for IP fragmentation.

TCP-Based Application: Determine TCP Connection Issues/Capabilities

Look for unsuccessful TCP handshakes. Examine the TCP handshake Options area, e.g. MSS, SACK.

UDP-Based Application: Identify Communication Issues

Look for unsuccessful requests.

Spot Application Errors

Filter for application error response codes.

Trace Files used in Topics from 89 to 172 are available at <http://www.wiresharkbook.com/troubleshooting.html>.

Topic 89: Wireshark Lab 1

This topic describes how to create a Troubleshooting Profile in Wireshark.

Until you create a new profile, you are working in Wireshark's Default profile. The profile you are working in is shown in the right side column of the Status Bar. This is shown next.

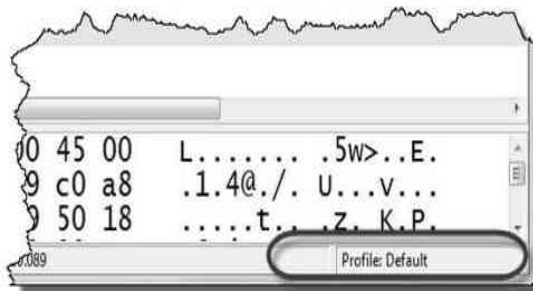


Figure 89.1: Create Your Troubleshooting Profile

You can create profiles to customize Wireshark with buttons, colors, and more. You can create separate profiles for different needs. For example, you may want to make a VoIP profile, a WLAN profile, and a general troubleshooting profile. You can quickly switch between profiles depending on your needs.

Step1: Right-click the **Profile** column on the Status Bar.

Step2: In the Configuration Profile window, select **New**.

Step3: Click the **arrow** in the **Create from** area, expand the **Global section** and select **Classic**. This profile uses the most vibrant colors.

Step4: Enter **Troubleshooting Book Profile** in the Profile Name area. Click **OK**.

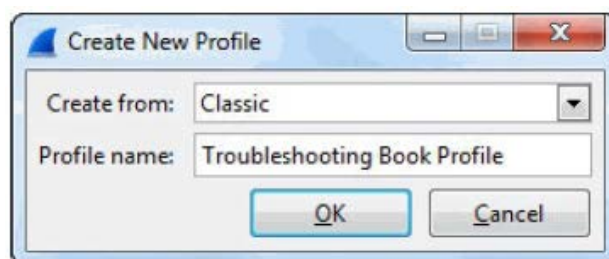


Figure 89.2: Troubleshooting Book Profile

As soon as you create your new profile, the Wireshark Status Bar indicates that you are working in the Troubleshooting Book Profile, as shown next.

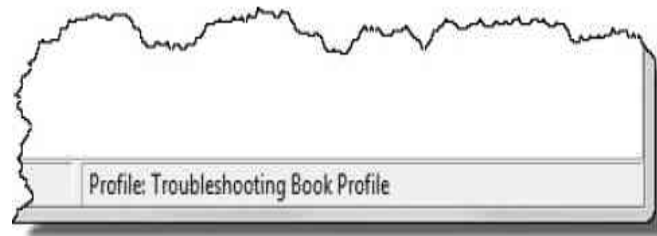


Figure 89.3: Wireshark Status Bar indicates the Troubleshooting Book Profile

You will be able to add capabilities and customization to this new profile. Wireshark also allows download/import a predefined profile for immediate use.

Topic 90: Wireshark Lab 2

This topic discusses how to enhance the Packet List Pane Columns in Wireshark.

By default, the Packet List pane contains: No. (number), Time, Source, Destination, Protocol, Length, and Info columns. This is shown next.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	24.6.173.220	96.17.148.114	TCP	66	523
2	0.017118000	96.17.148.114	24.6.173.220	TCP	66	80-
3	0.017290000	24.6.173.220	96.17.148.114	TCP	54	523

Figure 90.1: Enhance the Packet List Pane Columns

You can add columns to display additional information about packets to speed up your analysis process.

Step 1: Open *tr-httpdelta.pcapng*. This trace file contains traffic to/from a user's machine that is checking for Windows updates as well as virus detection updates.

Step 2: Packets 1-3 are TCP handshake packets. Packet 4 is an HTTP GET request for a file called *minitri.flg*. Packet 5 is an ACK for GET request. Packet 6 is the HTTP 200 OK. This is shown next.

1	0.000000000	24.6.173.220	96.17.148.114	TCP	66 52382-80 [SYN] Seq=0
2	0.017118000	96.17.148.114	24.6.173.220	TCP	66 80-52382 [SYN, ACK]
3	0.017290000	24.6.173.220	96.17.148.114	TCP	54 52382-80 [ACK] Seq=1
4	0.018186000	24.6.173.220	96.17.148.114	HTTP	343 GET /minitri.flg HTTP/1.1
5	0.036422000	96.17.148.114	24.6.173.220	TCP	60 80-52382 [ACK] Seq=1
6	0.037222000	96.17.148.114	24.6.173.220	HTTP	317 HTTP/1.1 200 OK (text/css)

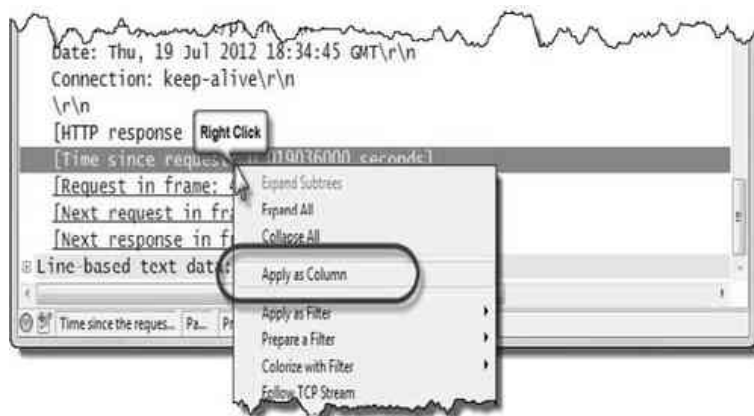
Figure 90.2: Enhance the Packet List Pane Columns

Select Packet 6 in the Packet List pane and then, in the Packet Details pane, expand the Hypertext Transfer Protocol.

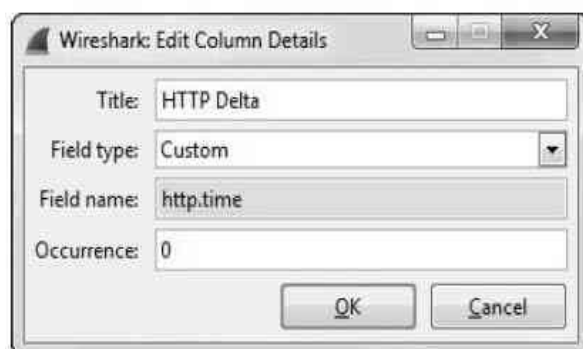


Figure 90.3: Add and Use a Custom Column to Locate HTTP Delays

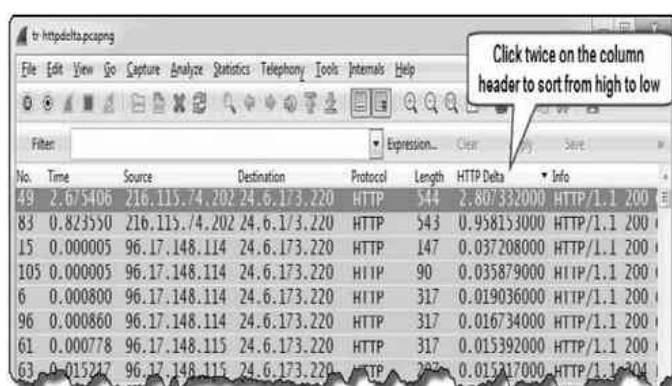
Step 3: Scroll to the bottom of the HTTP section and right-click on the [Time since request: 0.019036000 seconds] line. Select Apply as Column.



Step 4: Wireshark places the new Time since Request column to the left of the Info column. Right-click on this new column heading and select **Edit Column Details**. Enter **HTTP Delta** in the Title area. Click **OK**.



Step 5: Click twice on your new **HTTP Delta** column header to sort the column data from high to low. Wireshark indicates there is a 2.807332 second delay before one of the HTTP 200 OK responses (Packet 49).



Step 6: Right-click on your **HTTP Delta** column heading and select **Hide Column**. You can restore this hidden column at any time. Right-click on any column header, select **Displayed Columns** and select column to restore.

Topic 91: Wireshark Lab 3

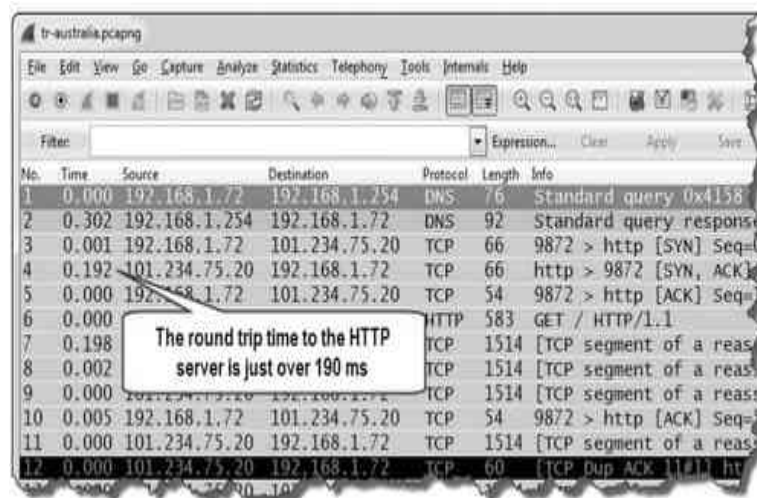
In this topic, we learn how to change the Time Column settings in Wireshark.

Packets are time stamped at the moment they are captured. By default, Wireshark sets the Time column to Seconds Since Beginning of Capture. In addition, the resolution is set to nanoseconds regardless of whether the packet timestamps contain that level of precision.

Step 1: Open *tr-australia.pcapng*. This trace file contains a web browsing session and was captured at the client. We want to change the Time column setting so we can quickly measure the delta time between displayed packets.

Step 2: This trace file begins with a DNS query and response. TCP connection establishment begins in Packet 3. Select **View | Time Display Format | Seconds since Previous Displayed Packet**.

Step 3: To change the precision, select **View | Time Display Format | Milliseconds: 0.123**. Now, look at the time from the SYN (Packet 3) to the SYN/ACK (Packet 4). It appears the round trip time is 192 (ms).



The image shows a Wireshark packet capture window titled 'tr-australia.pcapng'. A callout box points to the time difference between packet 3 and packet 4, stating: 'The round trip time to the HTTP server is just over 190 ms'.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000	192.168.1.72	192.168.1.254	DNS	76	Standard query 0x4158
2	0.302	192.168.1.254	192.168.1.72	DNS	92	Standard query response
3	0.001	192.168.1.72	101.234.75.20	TCP	66	9872 > http [SYN] Seq=0
4	0.192	101.234.75.20	192.168.1.72	TCP	66	http > 9872 [SYN, ACK]
5	0.000	192.168.1.72	101.234.75.20	TCP	54	9872 > http [ACK] Seq=
6	0.000			HTTP	583	GET / HTTP/1.1
7	0.198			TCP	1514	[TCP segment of a reas
8	0.002			TCP	1514	[TCP segment of a reas
9	0.000			TCP	1514	[TCP segment of a reas
10	0.005	192.168.1.72	101.234.75.20	TCP	54	9872 > http [ACK] Seq=
11	0.000	101.234.75.20	192.168.1.72	TCP	1514	[TCP segment of a reas
12	0.000	101.234.75.20	192.168.1.72	TCP	60	[TCP Dup ACK [1]] ht

Figure 91.1: Set the Time Column to Detect Path Latency

Topic 92: Wireshark Lab 4

This topic describes how to apply a filter on a host, subnet or conversation in Wireshark. When you capture traffic at the server or inside the network infrastructure: your trace file may contain conversations between many hosts on the network. Are you interested in the traffic between a specific client and server? If yes, you can apply a display filter based on a host address, a subnet address or a conversation.

Step 1: Open *tr-cnn.pcapng*.

Step 2: This trace file contains numerous conversations. Let's extract conversations between the local client and cnn.com servers. We begin with looking at the name resolution information that Wireshark extracted from the trace file. Select **Statistics | Show Address Resolution**.



The image shows the 'Address Resolution' window in Wireshark. It lists host information gathered from the trace file 'tr-cnn.pcapng'. The entry for 'sves.cnn.com' is circled.

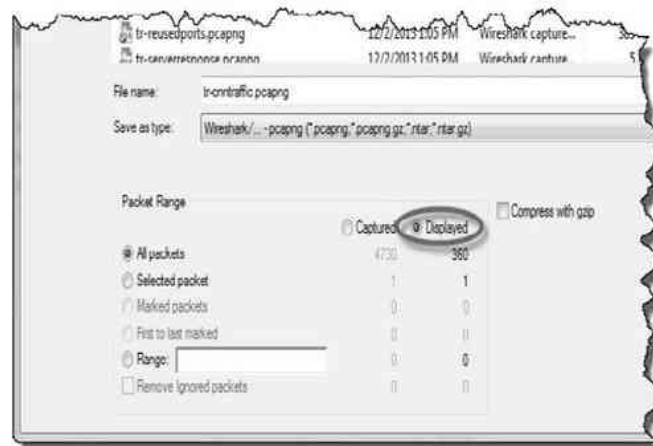
IP Address	Host Name
50.31.185.44	t1.visualrevenue.com
74.217.148.112	allides.outbrain.com
50.212.104.35	beacon-4-10-731352885.us-west-2.elb.amazonaws.com
157.166.224.32	sves.cnn.com
50.40.214.97	dualstack-109-334788911.us-east-1.elb.amazonaws.com
63.140.35.102	cnn.122.207.net
157.166.249.13	cnnintl-56m.gs1b.vgtf.net
157.166.249.11	cnn-56m.gs1b.vgtf.net
63.140.35.160	cnn.122.207.net

Figure 92.1: Extract and Save a Single Conversation

Servers in cnn.com domain all begin with 157.166. Click **OK** to close the Address Resolution window.

Step 3: In the display filter area, enter **ip.addr==157.166.0.0/16**. Click **Apply**. The Status Bar indicates that 360 packets match the filter. This lists all conversations to and from the cnn.com servers.

Step 4: Select **File | Export Specified Packets**. The **Displayed** radio button is selected by default, as shown next. Name your file **tr-cnntraffic.pcapng**. Click **Save**.



Step 5: We can also use right-click method to apply a display filter to a single conversation. Let's do this on **Packet 3** in the Packet List pane. Select **Conversation Filter | TCP**. 55 packets appear.

Step 6: Select **File | Export Specified Packets**. Name your file **tr-cnnconv1.pcapng**. Click **Save**.

Step 7: Click the **Clear** button to remove your display filter. Avoiding unrelated traffic and potential distractions makes analysis easier. Therefore, save the traffic from an interesting conversation to a separate trace file.

Topic 93: Wireshark Lab 5

This topic describes how to apply a filter using port number in Wireshark.

There are two ways to define a display filter on an application in a trace file—you can filter based on the application name (if known to Wireshark), or the port number in use.

Step 1: Open **tr-twohosts.pcapng**. Let's look at the FTP data transfer connection established by 192.168.1.119.

Step 2: Enter **ip.addr==192.168.1.119** in the display filter area and click **Apply**. This filters all traffic to and from 192.168.1.119. Look for response to the PASV command (Packet 3,959). Expand FTP section in the Packet Details pane to see port no. that the server will be listening on for the FTP data channel (port 39,757).

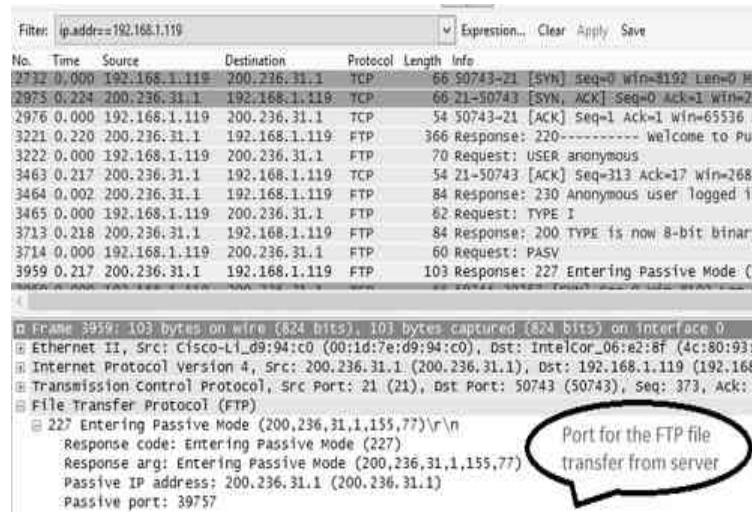
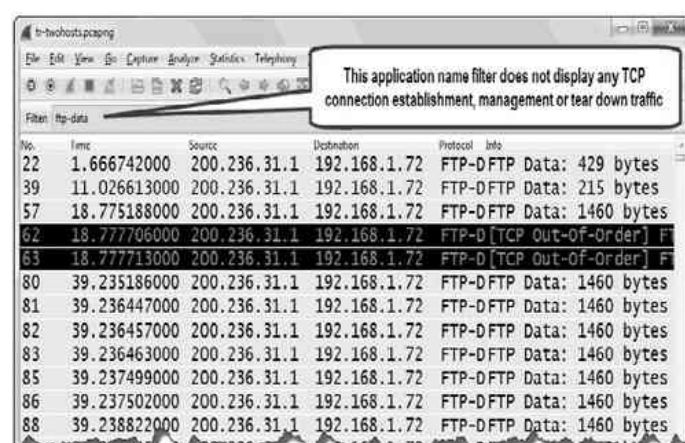


Figure 93.1: Filter Traffic Based on a Port Number

Step 3: Replace your address display filter with **tcp.port==39757**. Click **Apply**. The Status Bar indicates that 28,020 packets match this filter. Next, we contrast this with a display filter based on an application name.

Step 4: Replace your TCP port filter with **ftp-data** and click **Apply**. Now you won't see the TCP handshake, ACKs, or the connection teardown packets.



Step 5: Click the **Clear** button to remove your display filter.

Topic 94: Wireshark Lab 6

In this topic, we apply a filter using a field value in Wireshark.

We often need to identify packets that contain a specific field or a specific field value. For example, the display filter **http.request.method** can be used to view all HTTP client request packets using **http.request.method** field. **dns.flags.rcode > 0** is used to identify DNS error responses with code greater than 0. If you have a packet that contains the field name in which you are interested, you can right-click on the field and select **Apply as Filter**.

Step 1: Open *tr-winsize.pcapng*, HTTP clients send commands such as GET and POST in the HTTP Request Method field. Let's display all such packets.

Step 2: Right-click on the **Hypertext Transfer Protocol** section in the Packet Details pane in **Packet 4** and select **Expand Subtrees**. When you click on the **Request Method: GET** line, the Status Bar provides the name of this field—**http.request.method**.

Step 3: Type **http.request.method** in the display filter area and click **Apply**. One packet matches the filter. We can use this filter to determine how many HTTP requests were sent to a server. For example, **ip.addr==10.1.1.1 && http.request.method** would display all HTTP requests to or from 10.1.1.1. If 10.1.1.1 is a client, you would only see HTTP requests sent from this host.

Step 4: Click the **Clear** button to remove the display filter.

Topic 95: Wireshark Lab 7

This topic describes how to locate Buffer Problems using the Calculated Window Size Field in Wireshark.

Every TCP packet sent by a host contains that host's available receive buffer space in the Window Size field. If Window Scaling is in use, this field value is multiplied by a scaling factor. When the advertised buffer space becomes equal to zero, the host cannot accept any more data—a Zero Window condition. Even a low Window Size value can stop a TCP peer from transmitting data.

Step 1: Open *tr-winsize.pcapng*.

Step 2: Expand any **TCP header** in the Packet Details pane. Right-click on the **Calculated window size** field and select **Prepare a Filter | Selected**.

Step 3: Change the display filter value to **tcp.window_size < 1000** and click **Apply**. This displays Packet 374 in which the client is advertising a 536-byte receive buffer. What will happen if the TCP peer has more than 536 bytes of data queued to transmit? This low Window size value will stop data transmission. Next, we look at this phenomenon.

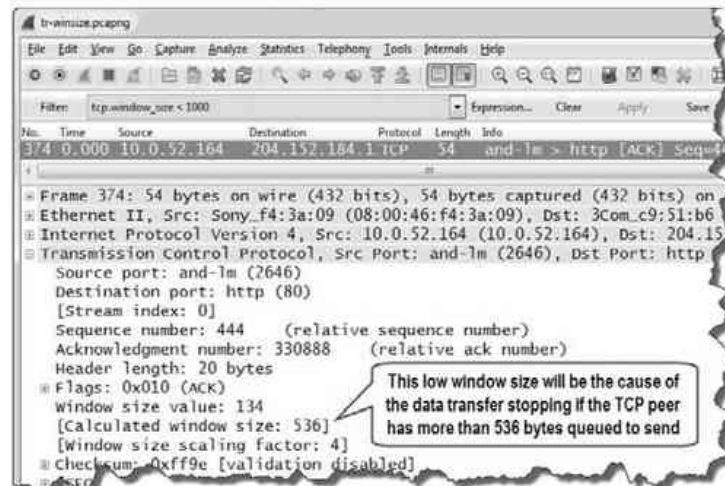
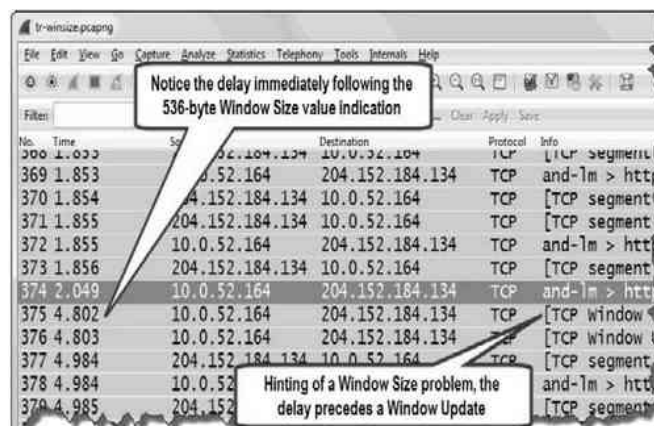


Figure 95.1: Filter on the Calculated Window Size Field to Locate Buffer Problems

Step 4: Click Clear to remove your filter. In this trace file, we next view the delay that occurs before the Window Update packet (Packet 375).



Essentially, the server could not transmit the full-sized packet because the client only had 536 bytes of buffer space available. The server had to wait until the client's buffer size increased.

Topic 96: Wireshark Lab 8

In this topic, we discuss how to filter OUT “Normal” Traffic to focus on anomalies in Wireshark.

To filter out traffic based on an application name, simply precede the application display filter name with an exclamation sign (!). For example, to remove ARP from view, use **!arp**. There are two methods you can use to exclude traffic based on a field value. One method uses the **!** or **not** operator with **==** or **eq**. The other uses the **!=** or **ne** operator.

Examples

- **!ip.addr==10.10.10.10**
- **http.request.method != "GET"**

When to Use ! and == and When to Use !=

Use the first method when you filter on a field name that matches two fields such as **ip.addr**, **tcp.port**, or **udp.port**. Use the second method when you refer to a field name that only matches one field such as **dns.flags.rcode** or **tcp.dstport**.

Correct Display Filter

- **!ip.addr==10.1.1.1**
- **dns.flags.rcode!=0**

Incorrect Display Filter

- **ip.addr != 10.1.1.1**
- **!dns.flags.rcode==0**

This lab removes a set of applications and protocols from view to determine what other traffic is seen on this network.

Step 1: Open *tr-general.pcapng*.

Step 2: In the display filter area, type **!tcp && !arp** and click **Apply**. There are 40 packets that match this filter.

Step 3: To remove DNS and DHCP from view, expand the filter by adding **&& !dns && !bootp**. Eight packets are displayed.

The displayed packets indicate that there are two hosts running Dropbox on the network and are sending packets to the broadcast address (255.255.255.255).

Topic 97: Wireshark Lab 9

This topic presents how to Create Filter Expression Buttons in Wireshark.

Filter Expression buttons are based on display filters. These buttons can be created and used to quickly apply display filters to your traffic to identify common network problems. Let's create a button to quickly identify TCP handshake packets that do not offer Selective Acknowledgment (SACK) or Window Scaling functionality. To do this, we need to combine an inclusion filter for the SYN bit set to 1 with an exclusion filter for the SACK and Window Scaling options.

Step 1: Open *tr-smbjoindomain.pcapng*.

Step 2: Packet 11 is the first SYN packet in the trace file. Right-click on the TCP header of this packet and select Expand Subtrees. Let's first build the filter and then turn the filter into a filter expression button. Right-click on **SYN: Set** line and select **Prepare a Filter | Selected**. Wireshark places first part of filter in the display filter area.

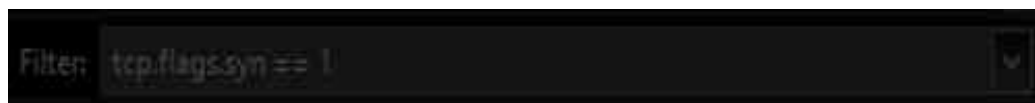


Figure 97.1: Create a Button to Detect Missing TCP Functionality

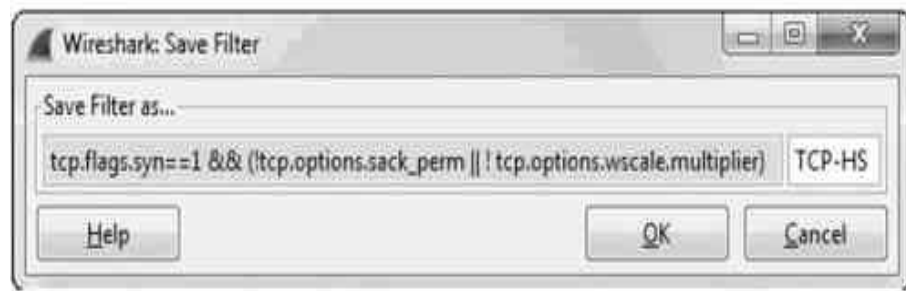
Step 3: Scroll down to the TCP Options area. Click on the **TCP SACK Permitted Option: True** line. This displays **tcp.options.sack_perm** in the Status Bar area. Expand the filter by typing **&& !tcp.options.sack_perm** so that TCP handshake packets that do not contain this value can be observed.



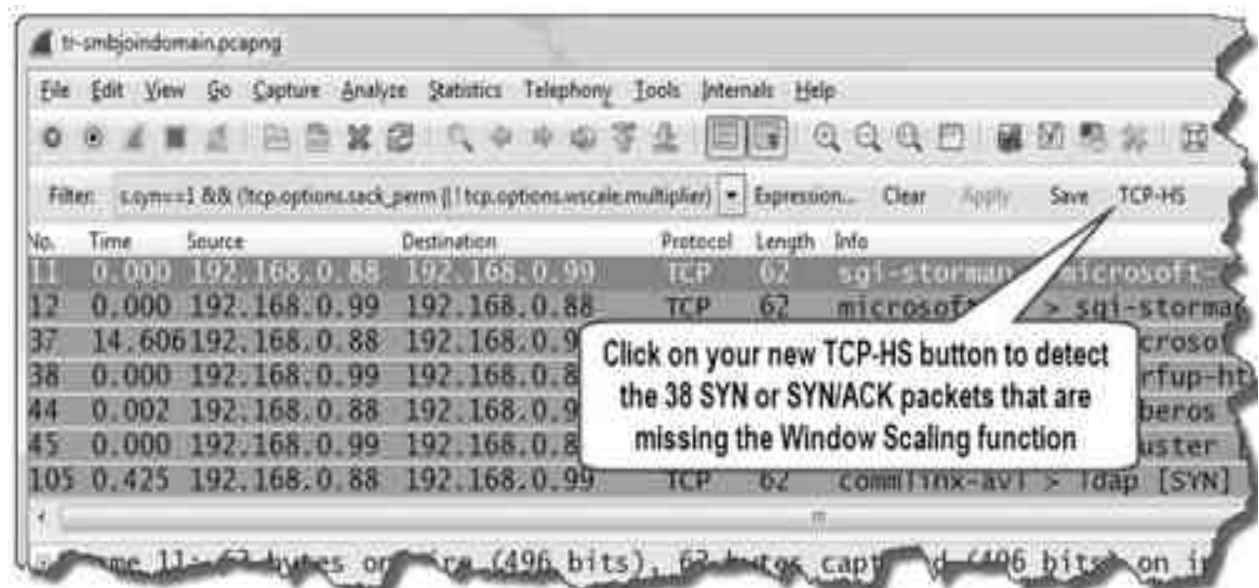
Step 4: Now add **|| !tcp.options.wscale.multiplier** and put parentheses around the options, as shown next.



Step 5: Click the **Save** button and name your new button **TCP-HS** and click **OK**



Step 6: Click your new TCP-HS button. There are 38 packets that match the TCP-HS button filter. The connections established by these packets will not support all the desired TCP functions.



If you need to edit, reorder or delete Filter Expression buttons, select **Edit | Preferences | Filter Expressions**.

Topic 98: Wireshark Lab 10

This topic discusses how to launch and navigate through the Expert Infos. in Wireshark.

Wireshark's Expert Infos can help you quickly detect network problems as well as obtain basic information about network communications and view/jump to packet comments.

Step 1: Open *tr-twohosts.pcapng*.

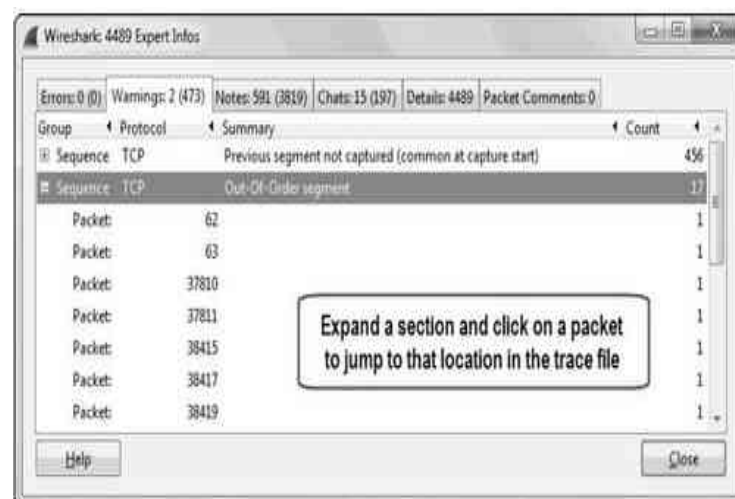
Step 2: Click the **Expert Infos** button in the bottom left corner of the Status Bar. This will open the **Expert Infos** window.



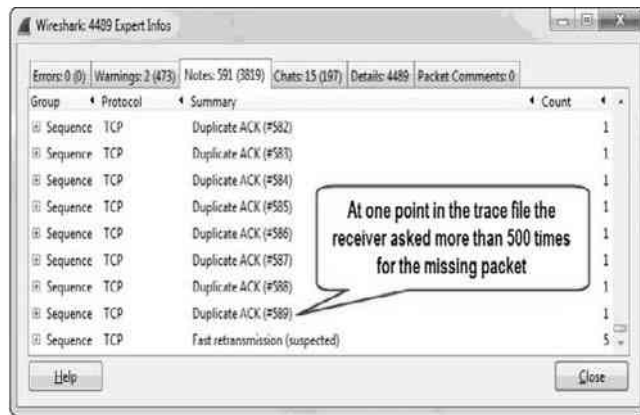
Figure 98.1: Use Expert Infos to Identify Network Problems

The Expert Infos window is divided into six tabs: **Errors**: Checksum errors, dissector failures. **Warnings**: Potential problems detected. **Notes**: Symptoms of problems; typically recovery processes. **Chats**: TCP connection overhead (handshake, Window updates, disconnects). **Details**: Summary of Errors, Warnings, Notes and Chats. **Packet Comments**: List of all packet comments in trace file.

Step 3: If IPv4 checksum validation is disabled, there are no Expert Infos Errors in this trace file. Otherwise, you will see 6,767 IPv4 Bad Checksum Errors. To disable IPv4 checksum validation, toggle back to Wireshark, rightclick on the **Internet Protocol Version 4** line in the Packet Details Pane, select **Protocol Preferences** and toggle off the *Validate IPv4 checksum if possible*. Click the Warnings tab of **Expert Infos** window. Click on a packet to jump to that location in the trace file.



Step 4: Click the **Notes** tab. As you scroll the list, you will see 589 Duplicate ACKs sent by receiver to recover a missing packet. This can be caused by a very high latency path or a brief connection outage.



Step 5: Click **Close** to shut down the Expert Infos window.

Topic 99: Wireshark Lab 11

This topic describes how to change the Dissector Behavior in Wireshark.

Some of Wireshark's predefined preference settings are not ideal for troubleshooting. For example, the **Allow subdissector to reassemble TCP streams** preference setting.

Step 1: Open *tr-youtubebad.pcapng*.

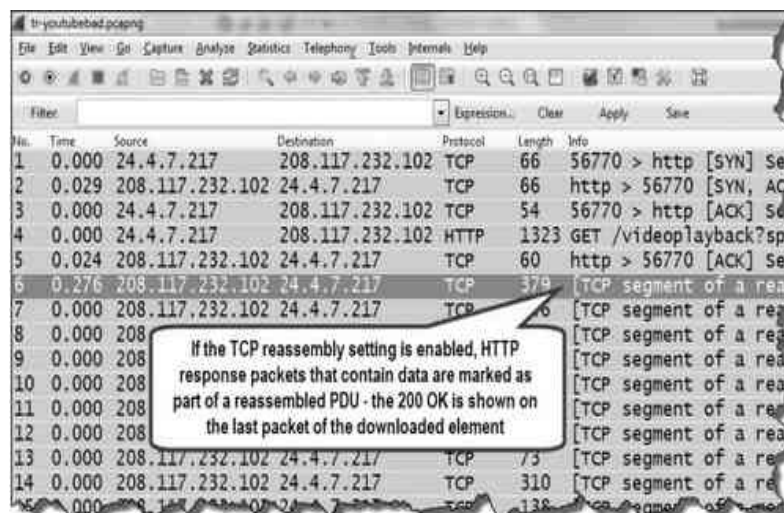


Figure 99.1: Change the TCP Dissector Reassembly Setting to Properly Measure HTTP Response Times

Response Code in the Info column for Packet 6 cannot be seen because the **Allow subdissector to reassemble TCP streams** preference setting is enabled.

Step 2: In the Packet Detail pane of **Packet 4**, right-click the **Hypertext Transfer Protocol** heading and select **Expand Subtrees**. A hyperlink to the response packet is

located at the bottom of the HTTP section. Double-click the **hyperlink** to jump to **Packet 29,259**.

```

Hypertext Transfer Protocol
[truncated] GET /videoplayback?spams=id%2Cexpire%2Cip%2Cipbits%2Citag
[truncated] Expert Info (Chat/Sequence): GET /videoplayback?spams=
[Message [truncated]: GET /videoplayback?spams=id%2Cexpire%2Cip%2
[Severity level: Chat]
[Group: Sequence]
Request Method: GET
Request URI [truncated]: /videoplayback?spams=id%2Cexpire%2Cip%2C
Request Version: HTTP/1.1
Host: v16.lscache8.c.youtube.com\r\n
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2.15
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
Keep-Alive: 115\r\n
Connection: keep-alive\r\n
[truncated] Cookie: VISITOR_INFO_LIVE=xU-BqDEfmck; use_hitbox=72c46ff
\r\n
[Full request URI [truncated]: http://v16.lscache8.c.youtube.com/video
[HTTP request 1/1]
[Response in frame: 29259]
  
```

Step 3: Scroll to the bottom of the HTTP header in Packet 29,259. The Time Since Request (**http.time**) field indicates the HTTP response time was over 276 seconds.

```

Hypertext Transfer Protocol
HTTP/1.1 200 OK\r\n
[truncated] Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n
[Message: HTTP/1.1 200 OK\r\n]
[Severity level: Chat]
[Group: Sequence]
Request Version: HTTP/1.1
Status Code: 200
Response Phrase: OK
Last-Modified: Sun, 13 Mar 2011 19:41:25 GMT\r\n
Content-Type: video/x-flv\r\n
Date: Mon, 14 Mar 2011 03:42:42 GMT\r\n
Expires: Mon, 14 Mar 2011 03:42:42 GMT\r\n
Cache-Control: private, max-age=22338\r\n
Accept-Ranges: bytes\r\n
Content-Length: 31095594\r\n
[Content length: 31095594]
Connection: close\r\n
X-Content-Type-Options: nosniff
Server: gvs 1.0\r\n
\r\n
[HTTP response 1/1]
[Time since request: 276.738716000 seconds]
[Request in frame: 4]
  
```

This is not correct. The HTTP response time is measured from the HTTP request packet to the HTTP response packet that contains the 200 OK response.

Step 4: To find the actual HTTP response time, in the Packet Details pane of any packet, right-click the **TCP header**, select **Protocol Preferences** and toggle off the **Allow subdissector to reassemble TCP streams** preference setting.

Step 5: Now click the Go to First Packet button. Notice we see that Packet 6 actually contains the 200 OK response. Examine the HTTP response time value in Packet 6. It is just over 300 ms.

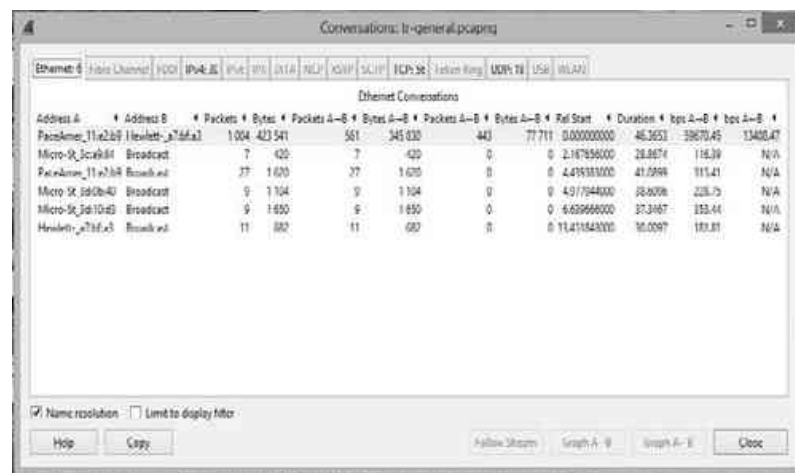
Topic 100: Wireshark Lab 12

This topic provides a way to find the top talkers in Wireshark.

We can use the Conversations window to determine the most active conversation based on hardware address, network address, or even the port numbers in use.

Step 1: Open *tr-general.pcapng*.

Step 2: Select **Statistics | Conversations**. The tabs indicate the number of each type of conversation seen in the trace file.

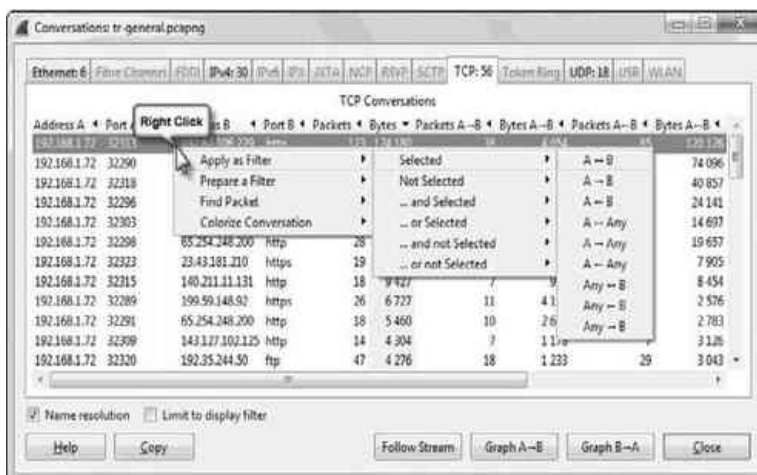


Address A	Address B	Packets	Bytes	Packets A-B	Bytes A-B	Packets B-A	Bytes B-A	Ref. Start	Duration	Seq. A-B	Seq. B-A
Packernet_11a2b3c	Headsnet_a7d4fa1	1004	423541	561	345830	443	77711	0.000000000	46.3653	3967045	1340047
Micro-St_Scan884	Broadcast	7	420	7	420	0	0	2.187650000	28.8674	116.39	N/A
Packernet_11a2b3c	Broadcast	27	1670	27	1670	0	0	4.419330000	41.5908	115.41	N/A
Micro-St_Scan884	Broadcast	9	1104	9	1104	0	0	4.917940000	28.6006	228.75	N/A
Micro-St_Scan884	Broadcast	9	1650	9	1650	0	0	6.639660000	37.3467	153.44	N/A
Headsnet_a7d4fa1	Broadcast	11	682	11	682	0	0	11.411640000	10.0097	181.81	N/A

Figure 100.1: Find the Most Active Conversation (Byte Count)

Step 3: We are interested in the most active TCP conversation (in bytes) in this trace file. Click the **TCP** tab and then click the **Bytes** column heading twice to sort from high to low. The most active conversation is between 192.168.1.72 on port 32313 and 192.87.106.229 on port 80 (listed as *http*).

Step 4: Right-click on this **top conversation** and select **Apply as Filter | Selected A <-> B**. Wireshark applies the filter and displays the 123 packets of this conversation.



Step 5: When you are done, click **Clear** to remove your display filter, toggle back to the Conversations window and click **Close**.

Topic 101: Wireshark Lab 13

This topic provides how to build a basic IO graph in Wireshark.

The basic IO Graph can be used to view throughput levels for all traffic in a trace file, or plot a subset of traffic (based on display filters). The IO Graph can help you prioritize your troubleshooting tasks.

Step 1: Open *tr-winsize.pcapng*. This trace file contains a single TCP conversation. An HTTP client is downloading a large file from a web server.

Step 2: Select **Statistics | IO Graph**. By default, Wireshark displays the packets per second rate (packet per tick with a default tick rate of one second).

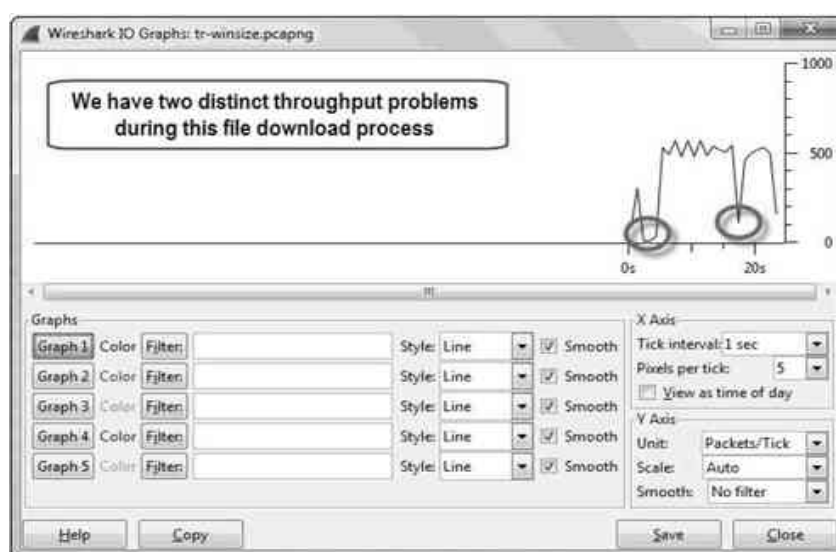
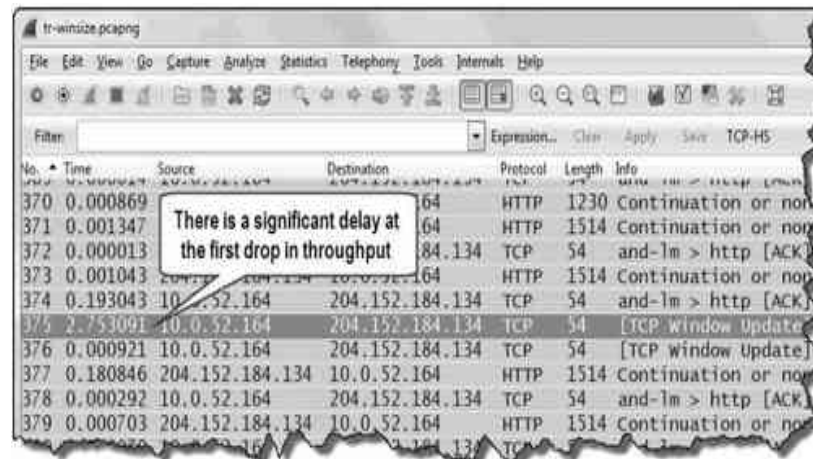


Figure 101.1: Quickly Spot a Throughput Problem in an IO Graph

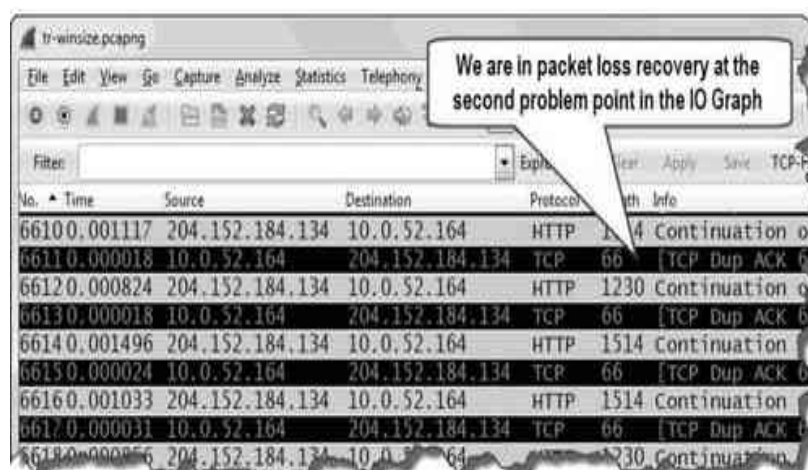
Let's investigate both of them.

Step 3: Click on the **first drop in throughput** in the graph. Wireshark jumps to that point in the trace file so we can investigate the problem further.



If your Time column is set to *Seconds Since Previous Displayed Packet* (**View | Time Display Format**), you will see a delay of over 2.75 seconds in the trace file. Packet 375 is marked as a TCP Window Update packet. This indicates that the delay may have something to do with the Window Size value advertised by the client (10.0.52.164).

Step 4: Go to the IO Graph. Click on the **second problem point** in the graph. From Wireshark's window, we can see what is happening at this point in the file download process.



The client appears to be in the middle of a packet loss recovery process.

Step 5: Go to the IO Graph and click **Close**.

Topic 102: Wireshark Lab 14

This topic discusses the addition of a coloring rule in Wireshark.

Coloring rules can be used to quickly identify packets in the Packet List pane. Let's build a coloring rule to highlight DNS errors.

Step 1: Open *tr-chappellu.pcapng*.

Step 2: Type **dns** in the display filter area to capture all DNS traffic and then click **Apply**. Forty-three packets will match your filter. The client asks two DNS servers (75.75.75.75 and 75.75.76.76) to resolve *www.chappellU.com* nine times before receiving an answer (Packet 12).

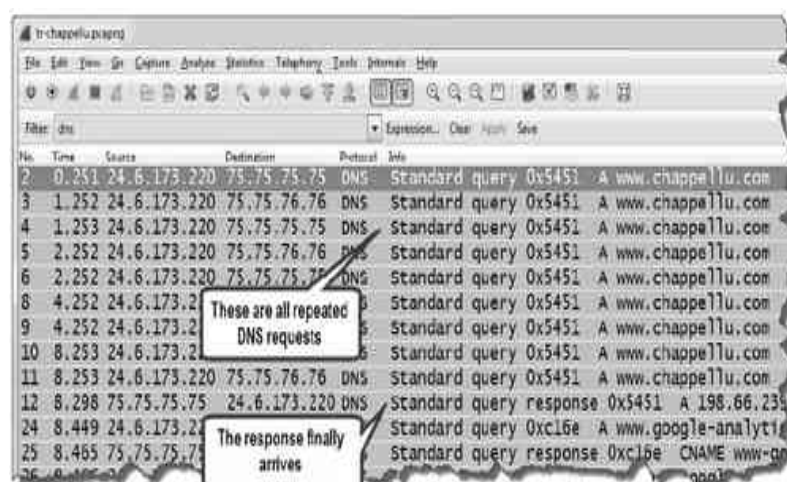


Figure 102.1: Build a Coloring Rule to Highlight DNS Errors

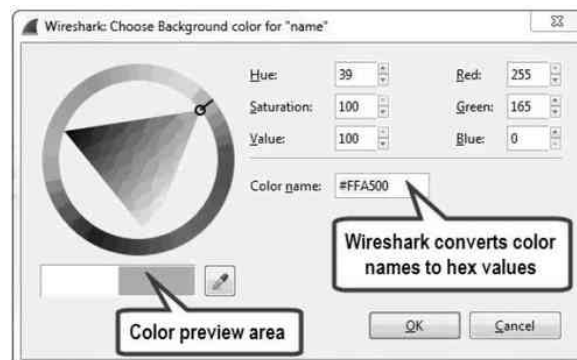
Step 3: Right-click on the **Domain Name System (query)** section in the Packet Details window of Packet 12. Select **Expand Subtrees**. When the Reply Code field inside the Flags section contains a 0, the DNS response was successful. If it contains any other value, the response indicates there is as DNS error.

Step 4: Right-click on the **Reply Code** field and select **Colorize with Filter | New Coloring Rule**.

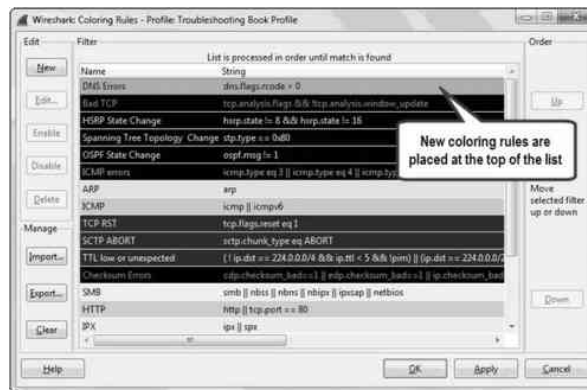
Step 5: Enter **DNS Errors** as the name of your new coloring rule. Change the String value to `dns.flags.rcode > 0`.



Step 6: Click the **Background Color** button and type **orange** in the Color Name field. When you tab away from the Color Name field, Wireshark changes the word "orange" to the hex value #FFA500. Then, Wireshark shows the new color in the color preview area. Click **OK** to close the background color window and click **OK** to close the Edit Color Filter window.

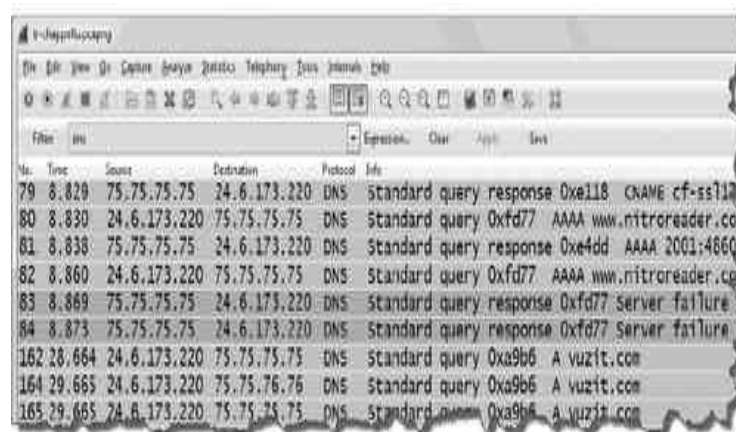


Your new coloring rule appears at the top of the list of color filters. Packets are processed in order through this list. Now, DNS errors will appear with an orange background.



Step 7: Click **OK** to close the Coloring Rules window.

Step 8: With **dns** filter still in place, scroll through the packets to see if you notice the two DNS errors in the trace file. Packets 83 & 84 appear with orange.



Step 9: Expand the **Frame** section in the Packet Details pane of Packet 83. You will see your Coloring Rule Name and Coloring Rule String listed in this area.

Topic 103: Capture Options for a Switched Network

This topic explains tips on choosing a capture location for a switched network.

These days, almost all network clients are connected to the network through a switch. Four types of packets are forwarded by a switch. Broadcast multicast, traffic to your hardware address and traffic to an unknown hardware address.

Because of this, when you connect a system running Wireshark directly to a switch port, you cannot listen to other users' traffic. In order to capture the traffic between the client and the upstream switch (and ultimately a remote host), you need to either

- (a) Install Wireshark or another capture tool on the user's machine,
- (b) Make the switch send a copy of the traffic down your analyzer port, or
- (c) Tap in and obtain a copy of the traffic between the client and the switch.

Install Wireshark on User's Machine: This is a great option—if you possible.

Switch Port Spanning: The next option to consider is to make the switch send a copy of the traffic down to your analyzer port (aka "spanning"). Not all switches support this feature. If there are corrupt frames traveling from the user's host, those corrupt frames won't be forwarded down the spanned port by the switch. Furthermore, avoid "oversubscribing" your switch port – spanning a level of traffic that cannot "fit" down the pipe to your Wireshark system. Switch will drop excess packets and trace file will be incomplete.

Use a Test Access Port ("Tap"): A tap is a simple device that copies all the traffic flowing through it (including those corrupt packets) out to a monitor port.

Topic 104: Wireshark Lab 15

In this topic, we describe the Wireless Capture Options in Wireshark.

You are lucky if your native WLAN adapter can capture WLAN Management and Control traffic. In Monitor Mode, you should be able to see traffic from any network as well. To test the capture capabilities of your WLAN Native Adapter, follow the given steps.

Step 1: Launch Wireshark and click the **Interface List** button on the Main Toolbar.

Step 2: Select the checkbox in front of your WLAN adapter and click **Start**.

Step 3: Go to a browser window and visit www.wireshark.org. Go to Wireshark and examine the packets you have captured. If your native adapter is suitable for network capture, you should see some WLAN management and Control traffic (such as Beacon packets and Probe Request/Probe Response packets).

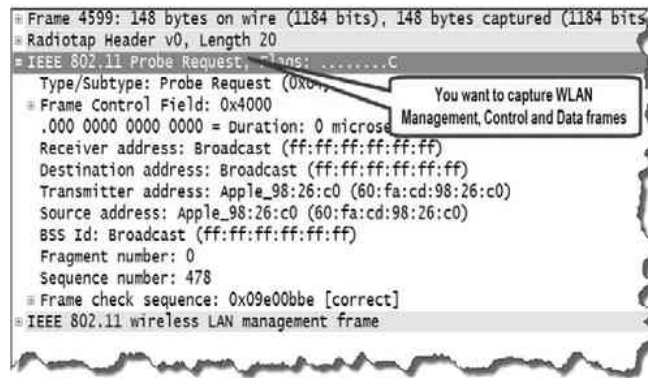


Figure 104.1: Test Your WLAN Native Adapter Capture Capabilities

Also, when you look at the data packets you should see an 802.11 header on the data packets. If your adapter strips off the 802.11 header, Wireshark will apply an Ethernet header. In case, you do not see these traffic types or characteristics, consider another solution for WLAN capture.

Visit: wiki.wireshark.org/CaptureSetup/WLAN for additional options for WLAN capture.

Topic 105: Wireshark Lab 16

This topic provides how to capture to a File Set in High Traffic Rate Situations in Wireshark.

Capturing to file sets is an important task when you are working in high traffic situations. File sets are groups of trace files that are linked based on their file name. In this lab exercise, we will use an autostop condition to only capture three files.

Step 1: Click the **Capture Options** button on the Main Toolbar.

Step 2: In the Capture Options window, set **path and file name** for your file set. Enable **Use Multiple Files**. Set **Next file every 1 Minute(s)**. Set **Stop capture after 3 Files**. Click **Start**.

Step 3: Open a browser and visit several web sites. Browse for at least **3 minutes** and go to Wireshark. Your capture process will automatically stop after 3 minutes. The third file will be displayed.

Step 4: To move from one file to the next file in a file set, select **File | File Set | List Files**, select a file in the list and Wireshark will load that file. You can quickly locate specific packets in the file set by applying a display filter to one of the files and then clicking subsequent files in the file list. The display filter will remain in place as you open each file. When you are finished navigating through files, click **Close** on the File Set window.

Topic 106: Wireshark Lab 17

This topic describes how to create and apply a MAC address filter in Wireshark.

Capture filters can reduce the traffic that you need to examine. Here, we will create and use a capture filter based on the MAC address. This will enable us to see all of the traffic to or from our machine.

Step 1: Obtain the MAC address of your host using either *ipconfig* or *ifconfig* as supported by your machine's OS.

Step 2: Click the **Capture Options** button on the Main Toolbar.

Step 3: Enter **ether host xx:xx:xx:xx:xx:xx** (replacing the x indications with your MAC address). Uncheck **Use multiple files**. Click **Start**. If you need this filter again, then click the **Capture Filter** button. Click **New** and name your filter **MyMac** and click **OK**.

Step 4: Open a browser window and visit www.wireshark.org.

Step 5: Go to Wireshark and click the **Stop Capture** button on the Main Toolbar.

Step 6: Your trace file will contain the HTTP traffic from your browsing session to www.wireshark.org.

Step 7: Clear the display filter when you are finished.

Topic 107: Verify the Target Host Traffic

This topic describes how to verify the capture process in Wireshark.

The first step of any analysis process is to verify that the hosts are able to communicate and you can see their traffic in the trace file. If hosts are not able to communicate, there can be several reasons for this.

Check Your Capture Process: If you do not see any traffic, something may have gone wrong during the capture process. A) Wireshark had a capture filter in place and the complaining user's traffic did not match the capture filter. B) If you are using a tap, the user's network cable isn't plugged into the tap. C) If you are spanning a switch port, the switch is not spanning the traffic down your analyzer port.

Consider the TCP/IP Resolution Process: All applications must go through a basic resolution process to build the packet to communicate with another host on a TCP/IP network.

1. Port Resolution: The user specifies Port number in the URL, in the application code.

2. Name Resolution: Examine local cache to locate the information or send a name resolution request on the network.

3. Location Resolution – Local or Remote: Compare target address to local host subnet mask.

4. MAC Address Resolution – Local Target: Examine local cache to locate the target MAC address or send an ARP Request.

5. Route Resolution: Examine local cache to locate the best route for target host.

6. MAC Address Resolution – Remote Target: Examine local cache to locate the target MAC address or send an ARP Request.

Topic 108: Wireshark Lab 18

This topic describes how to identify name Resolution Problems in Wireshark.

If a client does not know the IP address of a target (either in cache or a local hosts file), the client can send a DNS query to obtain this information. What if the client does not know of a DNS server to ask, the client cannot send out a DNS query. The resolution process ends there. If a client sends a DNS query and does not get a response or receives an error in the DNS response, the client can't talk to the target.

Step 1: Open *tr-nameresolution.pcapng*.

Step 2: Type **dns** in the display filter area and click **Apply**. You can observe from the Status Bar that 32 packets match this filter. From the Info column, you can see a number of **No Such Name** responses indicating the name resolution process failed. This can be due to DNS issues or user issues. In this trace file, the user typed the wrong URL.

Step 3: Clear the display filter when you are finished.

Topic 109: Wireshark Lab 19

This topic describes how to identify local address Resolution Problem in Wireshark.

Before a client can send a packet to a local target or a local router, it must obtain the MAC (Media Access Control) address of that local target or router. If the client does not have the MAC address information in cache, the client sends out an Address Resolution Protocol (ARP) request. If no response is received when trying to acquire the local target's MAC address, the client is done. It cannot send out a packet to the target.

Step 1: Open *tr-localresolution.pcapng*.

Step 2: Scroll through this trace file. Look at the ARP requests sent to discover the MAC address of 192.168.1.45. There are no responses.

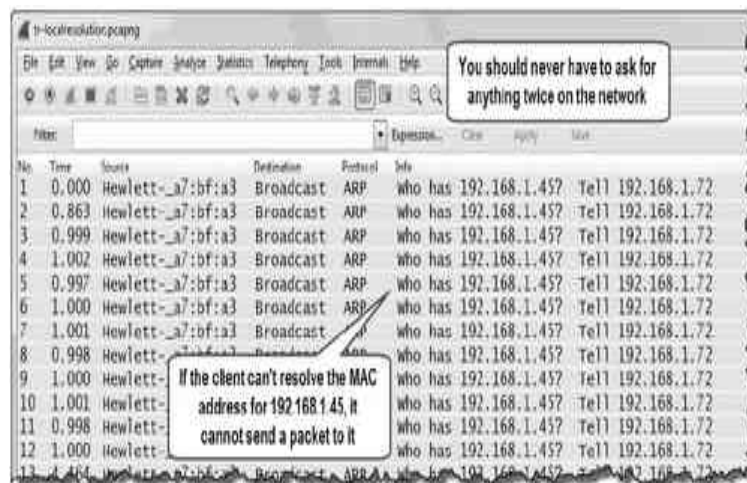


Figure 109.1: Find Local Address Resolution Problems

Without resolving MAC address issue, 192.168.1.72 will not be able to communicate with 192.168.1.45. Maybe 192.168.1.45 is not currently up. Maybe that is the wrong IP address. We can't tell.

Topic 110: Wireshark Lab 20

In this topic, we describe the “No server Response to TCP Connection Request” situation in Wireshark.

There are several reasons why a server may not respond to a TCP connection attempt. The TCP handshake request packet (SYN) may not arrive at the server. Maybe the SYN packet was lost. A firewall along the path dropped the SYN packet, or a host-based firewall on the server blocked access to the port. Alternately, the response (SYN/ACK) may not arrive at the client. The SYN/ACK packet was lost along the path, or a firewall along the path blocked the SYN/ACK.

Step 1: Open *tr-noserver.pcapng*.

Step 2: Scroll through this trace file. This trace file contains only SYN packets from 192.168.1.72 to 192.168.1.66. None of the SYN packets have received SYN/ACK responses. Expand the TCP header in Packet 1. Right-click on the **[Stream index: 0]** line and select **Apply as Column**. Wireshark shows that there are 6 separate connections (TCP Stream Index: 0-5).

No.	Time	Source	Destination	Protocol	Stream index	Source Port	Info
1	0.000	192.168.1.72	192.168.1.66	TCP	0	5538	5538 > http
2	2.042	192.168.1.72	192.168.1.66	TCP	1	5537	5537 > http
3	0.249	192.168.1.72	192.168.1.66	TCP	0	5538	[TCP Retran
4	5.750	192.168.1.72	192.168.1.66	TCP	1	5537	[TCP Retran
5	0.249	192.168.1.72	192.168.1.66	TCP	0	5538	[TCP Retran
6	11.327	192.168.1.72	192.168.1.66	TCP	2	5539	5539 > http
7	0.251	192.168.1.72	192.168.1.66	TCP	3	5540	5540 > http
8	0.425	192.168.1.72	192.168.1.66	TCP	4	5541	5541 > http
9	2.121	192.168.1.72	192.168.1.66	TCP	2	5539	[TCP Retran
10	0.257	192.168.1.72	192.168.1.66	TCP	3	5540	[TCP Retran
11	0.420	192.1			4	5541	[TCP Retran
12	5.314	192.1			2	5539	[TCP Retran
13	0.261	192.1			3	5540	[TCP Retran
14	0.424	192.168.1.72	192.168.1.66	TCP	4	5541	[TCP Retran
15	11.580	192.168.1.72	192.168.1.66	TCP	5	5545	5545 > http
16	2.999	192.168.1.72	192.168.1.66	TCP	5	5545	[TCP Retran
17	5.996	192.168.1.72	192.168.1.66	TCP	5	5545	[TCP Retran

Figure 110.1: No Response to TCP Connection Request

Step 3: Right-click on the TCP **Source Port** field in any packet and select **Apply as Column**. This shows that the client has set up numerous ports for these connections.

Step 4: Some of these SYN packets match the HTTP coloring rule while others match the Bad TCP coloring rule because the SYN packets are Retransmissions.

Topic 111: Wireshark Lab 21

This topic discusses how to analyze the “No Response to Service Request” situation in Wireshark.

Step 1: Open *tr-serverresponse.pcapng*.

Step 2: Scroll through this trace file to get familiar with the traffic pattern. Use the Stream Index column to differentiate the separate connections bet. 24.6.173.220 and 50.62.146.230. Right-click on Packet 1 in the Packet List pane and select **Conversation Filter | TCP**.

No.	Time	Source	Destination	Protocol	Stream index	Info
1	0.000	24.6.173.220	50.62.146.230	TCP	0	44043 > http [SYN Seq=1565847
2	0.035	50.62.146.230	24.6.173.220	TCP	0	http > 44043 [SYN, ACK] Seq=156
3	0.000	24.6.173.220	50.62.146.230	TCP	0	44043 > http [ACK Seq=1565847
4	0.000	24.6.173.220	50.62.146.230	HTTP	0	GET / HTTP/1.1
5	0.036	50.62.146.230	24.6.173.220	TCP	0	http > 44043 [ACK Seq=1588409
6	7.631	24.6.173.220	50.62.146.230	TCP	0	44043 > http [FIN, ACK] Seq=156
11	0.075	50.62.146.230	24.6.173.220	TCP	0	http > 44043 [ACK Seq=1588409
21	119.99	24.6.173.220	50.62.146.230	TCP	0	44043 > http [RST, ACK] Seq=156

Figure 111.1: No Response to Service Request

The TCP handshake completes successfully in Packets 1-3. The client requests the default file from the web site's root directory in Packet 4. Packet 5 contains acknowledgment no. 288, which indicates that the server has received every sequence no. up to 287, and it expects sequence no. 288 next. So, server received the request. Packet 5 contains acknowledgment no. 288, which indicates that the server has received every sequence no. up to 287, and it expects sequence no. 288 next. So, server received the request. The client's browser appears to time out and sends a FIN/ACK after almost 8 seconds. The client begins an implied connection termination. The server sends an ACK. We would expect the server to begin closing its side of the connection, but it does not. The client waits almost 120 seconds before sending a RST/ACK. TCP appears to be functioning properly in this trace file. The symptoms indicate the application has failed at the server side.

Topic 112: Do not Focus on Acceptable Delays

This topic discusses the worth of normal delays.

You can safely ignore some delays in your trace files. For example, a delay before a TCP RST packet would likely not be felt by the end user. Such acceptable delays should not raise an alarm.

Delays before DNS Queries

A DNS query will be triggered when we enter a URL in a browser and then press Enter. When we see an interesting link on a web page and we click on it. Open *tr-delays.pcapng*.

You will find that the delay before Packet 29 is caused by an eventual time out of a connection. Packet 32 is a DNS query and the delay before it is because the user did not straightaway click on www.wireshark.org.

Delays before TCP FIN or Reset Packets

Delay introduced by an application, which sends TCP FIN/RST packets to close the connection after waiting for specified time or some task is completed.

Delays before a Client Sends a Request to a Server

Filling out a form and pressing the Submit button, or clicking the next link on a web page. In all such cases, an application requires user interaction.

Delays before Keep-Alive or Zero Window Probes

Keep-Alives and Zero Window packets are sent during a Zero Window situation to determine if more buffer space is available at the target. Hosts also send “Keep-Alives” periodically to maintain a TCP connection.

Delays before TLS Encrypted Alert

An application sends a Transport Layer Security (TLS) Encrypted Alert to eventually close an encrypted connection; we can't see the Close command.

Delays before a Periodic Set of Packets in a Connection that is Otherwise Idle

Applications can define their own keep alive packet types to keep the connection active.

Topic 113: Watch for the Delays that DO Matter

This topic describes various delays that do matter for networks.

On your network, knowing what "normal" delay times are will help you identify unusually high delay times. We next provide a list of delays that should be examined.

Delays before a Server Responds with a SYN/ACK

A client can use the time between the SYN and SYN/ACK in the TCP handshake to determine the round trip time between the hosts. A large delay before the SYN/ACK is an indication of a high round trip time between the hosts.

Delays before a Client Completes the 3-Way TCP Handshake

Time between SYN/ACK and client's ACK to finish TCP handshake can be used by server to determine the round trip time.

Delays before a Server Sends a Response

In a trace file, if you observe that a server quickly sent an ACK to a client's request, but there is a long delay before server's response, then this is not path latency issue. We need to consider why a server might be slow. For example, let's look at the case given in *tr-http-pcapnet101.pcapng*. The client's GET request (GET /home) in packet 18 followed by the server ACK takes 17 ms. We can see a delay of almost 1.8 seconds before the requested data begins. The trace was taken at the client. The round trip time between the hosts is acceptable, but the server response time is not.

Delays before the Next Packet in a Data Stream

There can be several reasons for delays that occur during a file download or file upload process. The sender became busy with other processing. There can be a

device along the path, which may be buffering the data for higher priority traffic. Lack of receive buffer space at a receiver.

Delays before an ACK from a TCP peer

This situation arises when ACKs from a receiver get delayed because of the path latency or delayed ACK function. Delayed ACK timers are often set at 200 ms.

Delays before a Window Update

A host must wait for a Window Update before sending a packet, if it finds that Window Size advertised is too small to fit a full-sized data segment.

Topic 114: Wireshark Lab 22

In this topic, we discuss how to detect delays in UDP Conversations in Wireshark.

The User Datagram Protocol (UDP) is a connectionless transport layer protocol with an 8-byte header. Unlike TCP, UDP has no sequencing or acknowledgement capability. Delays between requests and responses are a measure for UDP-based applications. UDP conversation statistics such as packet rate, bps rate etc. can be obtained with Conversations Window.

Step 1: Open *tr-voip-extensions.pcapng*.

Step 2: Select Statistics | Conversation.

Step 3: Click the UDP tab. Uncheck the Name Resolution option, if you are interested in seeing port numbers rather than port names.

Step 4: Click twice on the **bps A → B** column heading. This will sort UDP conversations based on traffic flowing from Address A/Port A to Address B/Port B. In the list, Wireshark will view the conversation between 192.168.5.11/port 25426 and 192.158.5.10/port 8000 first.

Conversations: tr-voip-extension.pcapng

Ethernet: 2 | Fibre Channel | FDDI | **IPv4: 2** | IPv6 | IPX | JXTA | NCP | RSVP | SCTP | TCP

UDP

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A-B	Bytes A-B
192.168.5.11	25426	192.168.5.10	8000	575	117 426	289	
192.168.5.10	5060	192.168.5.11	5060	2	1 973	2	
192.168.5.11	25427	192.168.5.10	8001	5	542	5	
192.168.5.10	5060	192.168.5.11	1073	12	6 152	0	
192.168.5.11	7818	192.168.5.10	5060	8	4 123	0	
192.168.5.20	5060	192.168.5.11	1033	3	2 254	0	
192.168.5.11	7818	192.168.5.20	5060	4	2 471	0	

☐ Name resolution ☐ Limit to display filter

Help Copy

Figure 114.1: Obtain UDP Conversation Statistics and Filter on a UDP Conversation

Step 5: This conversation is also the most active UDP conversation in terms of the Bytes column value. Right-click on this conversation line and choose **Apply as Filter | Selected | A <=> B**.

Step 6: Select **File | Export Specified Packets** and name your new file **udpconv1.pcapng**. When you are done, clear the display filter and close the Conversations window. Removal of unrelated traffic from view, allows Wireshark to list down statistics related to that traffic set. This makes identification of performance issues easier.

Topic 115: Wireshark Lab 23

This topic describes how to Add/Sort a Delta Time Column in Wireshark.

In Wireshark, the default Time column setting is **Seconds Since Beginning of Capture**. It becomes easier to locate delays when a time column displays delta times. We can locate the largest delays in a trace file by sorting the delta time column.

Step 1: Open **tr-malaysianairlines.pcapng**

Step 2: Expand the Frame section of any packet.

Step 3: Right-click on the “Time delta from previous captured frame” line and select **Apply as Column**. This will create a **frame.time_delta** column.

Step 4: The new column appears to the left of the Info column. Click and drag it to the right of the existing Time column.

Step 5: Rename the column by right-clicking on the column heading and selecting **Edit Column Details**. Let’s change the Title to **Delta**. Click **OK**.

Step 6: Click your new Delta column heading twice to sort from high to low. Go to the first packet. You can see packets with the largest delays at the top of the list.

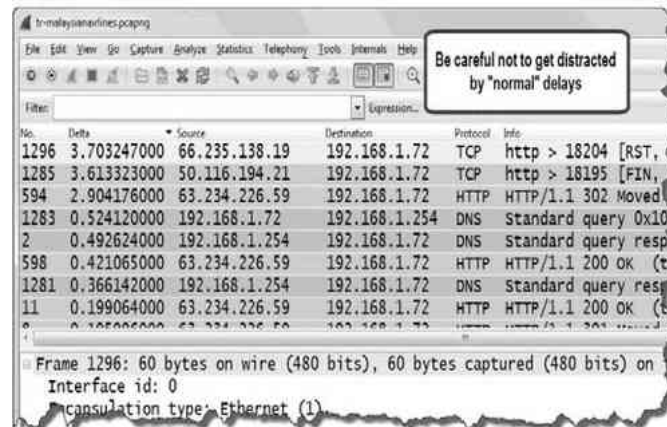


Figure 115.1: Add/Sort a Delta Time Column

In this trace file, delays such as delays before DNS queries, TCP RST packets, or TCP FIN packets are “normal”. We do not care about them. Focus on delays before DNS and HTTP responses.

Topic 116: Wireshark Lab 24

This topic describes how to add/sort a delta displayed time column in Wireshark.

Delays between displayed packets can be identified by using a delta displayed time column. Let’s create such a time column to show the delta times of DNS traffic only.

Step 1: Open *tr-malaysianairlines.pcapng*.

Step 2: Click the **Preferences** button on the Main Toolbar and then select Columns.

Step 3: Click the **Add** button. In the drop-down “Field Type” list, select **Delta time displayed**. Click on the column name and update the name to **Delta Displayed**.

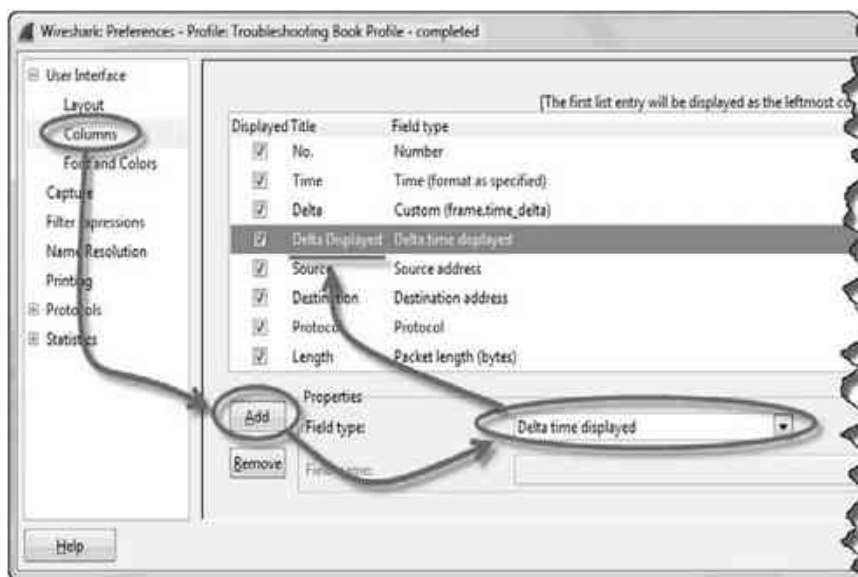


Figure 116.1: Add/Sort a Delta Displayed Time Column

Step 4: Click and drag your **Delta Displayed** column above the “Source” column. Click **OK**. If desired, right click on your newly created column heading to set left alignment.

Step 5: Now apply a filter for DNS traffic. For this purpose, enter **dns** in the display filter area. Click **Apply**.

Step 6: Clicking twice on your new **Delta Displayed** column will sort from high to low. Let’s not focus on delays before DNS queries, and care about delays before DNS query responses.

No.	Delta	Delta Displayed	Source	Destination	Length	Protocol	Info
942	0.0002584.294		192.168.1.72	192.168.1.25	88	DNS	Standard query 0x2f26 A 232
17	0.0010810.614		192.168.1.72	192.168.1.25	76	DNS	Standard query 0xf2dc A s3.
1283	0.5241200.524		192.168.1.72	192.168.1.25	78	DNS	Standard query 0x10c7 A www
2	0.4926240.492		192.168.1.254	192.168.1.72	306	UNR	Standard query response 0x8e
1229	0.0078310.396		192.168.1.72	192.168.1.25	76	DNS	Standard query 0x533f A www
1281	0.3661420.366		192.168.1.254	192.168.1.72	222	DNS	Standard query response 0xb0
365	0.0202050.359		192.168.1.72	192.168.1.25	88	DNS	Standard query 0xf1e6 A mail
410	0.0014790.306		192.168.1.72	192.168.1.25	89	DNS	Standard query 0xdbc1 A d3e
1139	0.0012880.256		192.168.1.72	192.168.1.25	88	DNS	Standard query 0x9c3b A mail
1757	0.0178620.735		192.168.1.254	192.168.1.72	104	DNS	Standard query response 0xbr
147	0.0010020.118		192.168.1.72	192.168.1.25	88	DNS	Standard query 0x16d7 A 232

We find that two packets (Packet 2 and Packet 1,281) are really very slow. **Cause for Delays:** If a local DNS server does not have these names in its cache, it needs to perform recursive queries to obtain the data.

Topic 117: Wireshark Lab 25

In this topic, we describe how to plot UDP Delays in Wireshark.

Using Wireshark's Advanced IO Graph with a filter and a reference to the maximum **frame.time_delta_displayed** value, we can have a pictorial idea about the delays in a trace file.

Step 1: Open *tr-queuing.pcapng*.

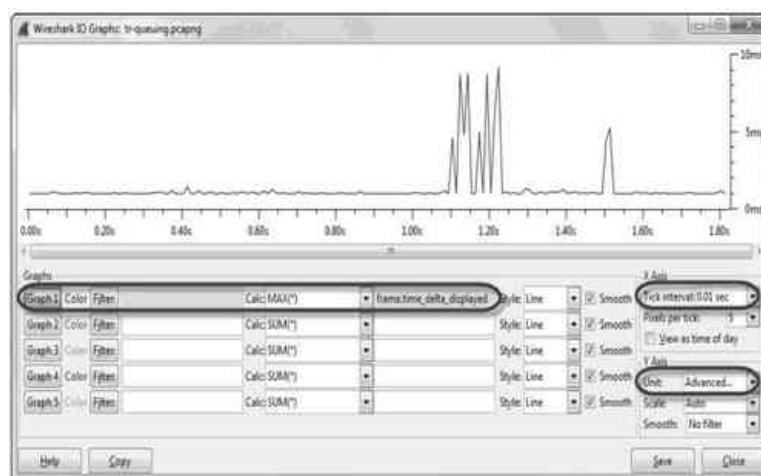
Step 2: Select **Statistics | IO Graph**.

Step 3: In the Y Axis Unit area, select **Advanced...**

Step 4: As this trace file contains less than 2 seconds of traffic. Select **0.01 sec**, in the X Axis Tick Interval area.

Step 5: Select the **MAX(*)** Graph 1 Calc option and enter **frame.time_delta_displayed** in the Calc area.

Step 6: Click the **Graph 1** button to plot the results. If your trace file contains both UDP and TCP-based traffics and you want to plot UDP delays, then enter **udp** in the Graph 1 filter area before you click the Graph 1 button.



There is a sudden increase in the delta time at approx. 1.2 seconds of the trace file. On clicking these points, Wireshark jumps to that point in the trace file and enables to do additional analysis.

Topic 118: Wireshark Lab 26

This topic describes how to obtain TCP conversation statistics in Wireshark.

Transmission Control Protocol (TCP) is a connection-oriented protocol. TCP is a stream based communication. Each separate TCP conversation is numbered by Wireshark with a TCP Stream Index (**tcp.stream**) value starting with 0. Let's find the

most active TCP conversation. If you have a large trace file, you can apply a filter from within Conversations window.

Step 1: Open *tr-chappellu.pcapng*.

Step 2: Select **Statistics | Conversations** and click the **TCP** tab.

Step 3: Click twice on the **Bytes** column to sort from high to low.

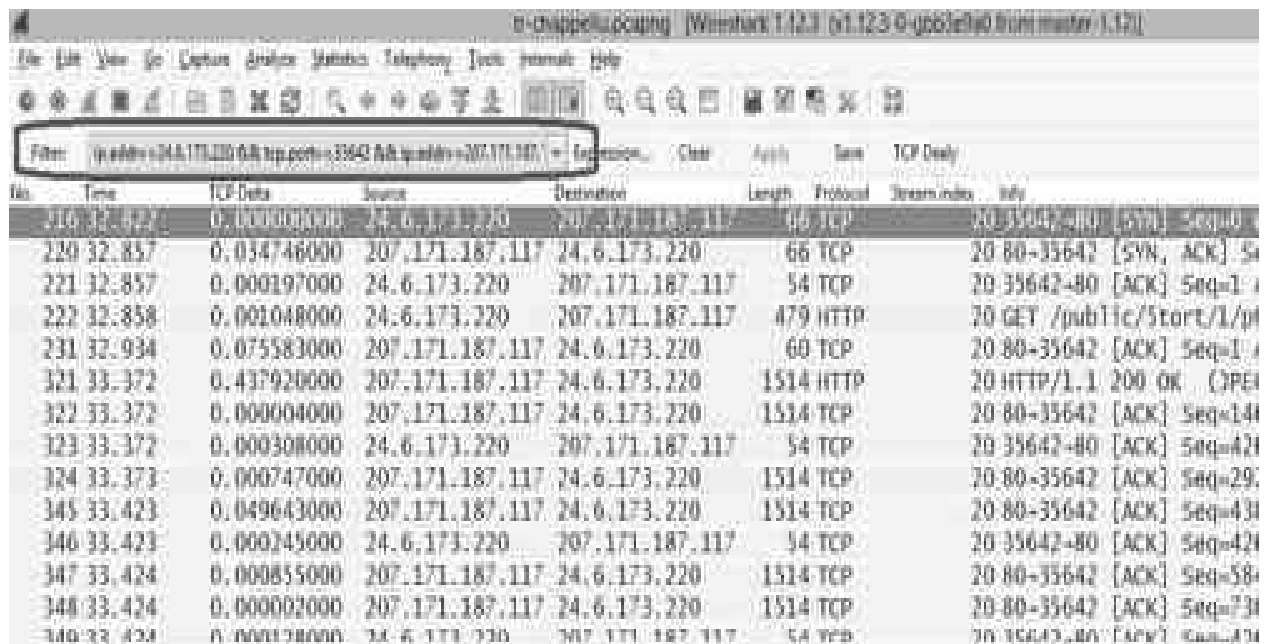
Address A	Port A	Address B	Port B	Packets	Bytes	Packets A-B
24.6.173.220	35642	207.171.187.117	http	127	126 605	41
24.6.173.220	35643	207.171.187.117	http	122	119 496	38
24.6.173.220	35641	207.171.187.117	http	85	85 381	27
24.6.173.220	35644	207.171.187.117	http	56	54 884	18
24.6.173.220	35640	207.171.187.117	http	54	49 399	18
24.6.173.220	35629	184.73.250.227	http	66	26 339	34
24.6.173.220	35623	69.59.180.202	http	22	12 078	1
24.6.173.220	35627	141.101.125.193	http	14	9 402	

Figure 118.1: Obtain TCP Conversation Statistics

Step 4: Right-click on the top entry and select **Apply as Filter | Selected | A <=> B**. Wireshark creates a filter based on the source/destination address and source/destination port fields.

TCP Conversations									
Address A	Port A	Address B	Port B	Packets	Bytes	Packets A-B	Bytes A-B	Packets A-B	Bytes A-B
24.6.173.220	35642	207.171.187.117	80	127	126 605	41	3 703		86
24.6.173.220	35643	207.171.187.117	80	122	119 496	38	3 015		84
24.6.173.220	35641	207.171.187.117	80	85	85 381	27	2 421		58
24.6.173.220	35644	207.171.187.117	80	56	54 884	18	1 409		38
24.6.173.220	35640	207.171.187.117	80	54	49 399	18	1 025		36

TCP Conversations									
A-B	Bytes A-B	Packets A-B	Bytes A-B	Rel Start	Duration	bps A-B	bps A-B		
41	3 703	86	122 902	32.822396000	1.5346	19303.65	640605.19		
38	3 015	84	116 481	32.827897000	1.1597	20798.30	803518.12		
27	2 421	58	82 960	32.613127000	1.2735	15208.74	521155.45		
18	1 409	38	53 475	32.860271000	1.2247	9204.26	349324.30		
18	1 025	36	47 464	32.546263000	1.1913	12994.01	318732.70		



The image shows a Wireshark packet capture window. The filter bar at the top contains the filter: `ip.addr==24.173.220.68 && tcp.port==35642 && stream.index==205171187`. The packet list below shows various TCP and HTTP packets. The 'Stream index' column is visible, and the filter is applied to it.

No.	Time	Length	Protocol	Stream index	Info
200	32.857	66	TCP	20-80-35642	[SYN, ACK] Seq=1...
221	32.857	54	TCP	20-35642-80	[ACK] Seq=1...
222	32.858	479	HTTP	20-80-35642	GET /public/Start/1/p...
231	32.934	60	TCP	20-80-35642	[ACK] Seq=1...
321	33.372	1514	HTTP	20-HTTP/1.1	200 OK (JPEG)
322	33.372	1514	TCP	20-80-35642	[ACK] Seq=14...
323	33.372	54	TCP	20-35642-80	[ACK] Seq=42...
324	33.373	1514	TCP	20-80-35642	[ACK] Seq=29...
345	33.423	1514	TCP	20-80-35642	[ACK] Seq=43...
346	33.423	54	TCP	20-35642-80	[ACK] Seq=42...
347	33.424	1514	TCP	20-80-35642	[ACK] Seq=58...
348	33.424	1514	TCP	20-80-35642	[ACK] Seq=73...
349	33.424	54	TCP	20-35642-80	[ACK] Seq=43...

Step 5: Click **Clear** to remove your filter when you are finished. If there are many TCP conversations contained in your trace file, the method we learnt in this topic can be used to find the most active conversation and then quickly apply a filter on that conversation.

Topic 119: Wireshark Lab 27

In this topic, we describe how to filter TCP conversation with the stream index field in Wireshark.

What is a Stream?

Wireshark assigns each unique TCP connection attempt a separate TCP stream based on the source/destination addresses and source/destination port numbers. Let's create a filter based on the **tcp.stream** field.

Step 1: Open *tr-chappellu.pcapng*.

Step 2: Expand the TCP header in Packet 59.

Step 3: Right-click on the **[Stream index: 7]** field in the TCP header. Select **Apply as Filter | Selected**.

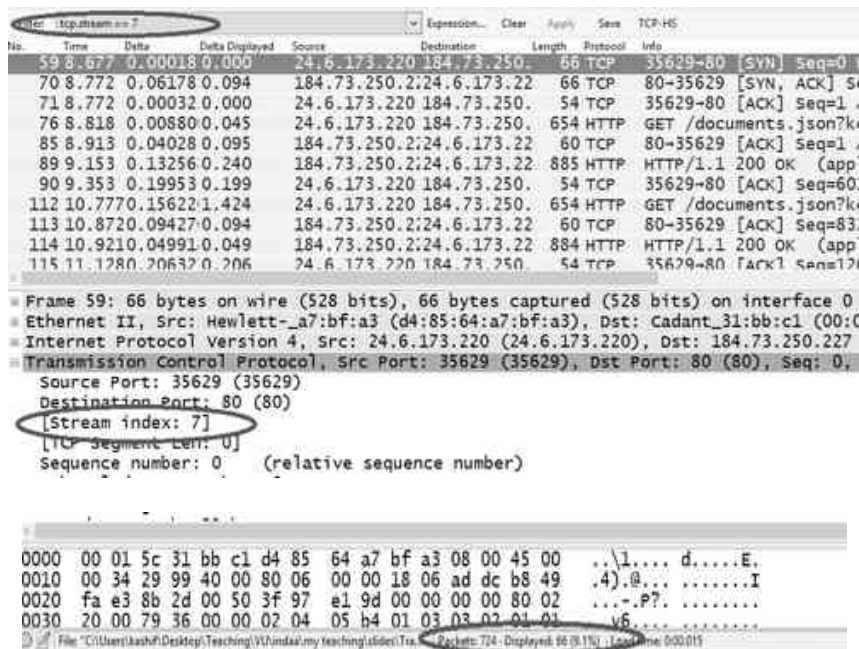


Figure 119.1: Filter on a TCP Conversation Using the Stream Index Field

We can see that Wireshark creates a filter for **tcp.stream==7** in the filter display area and applies it to the trace file. There are 66 packets matching this filter as indicated on the Status Bar. You can save this TCP conversation in a separate trace file. Select **File | Export Specified Packets** and provide a file name.

Step 4: Once you are finished, click **Clear** to remove the filter.

Topic 120: Wireshark Lab 28

This topic shows how to add a TCP stream index column in Wireshark.

There can be numerous TCP conversations in a trace file. Wireshark calls each unique connection attempt a TCP stream. To differentiate the TCP conversations, Wireshark assigns each unique connection attempt a separate TCP stream index based on the source/destination addresses and source/destination port numbers. Wireshark numbers each separate TCP conversation with a TCP Stream Index (**tcp.stream**) value starting with 0. Next, we show that packet, which is part of the first TCP conversation in a trace file.

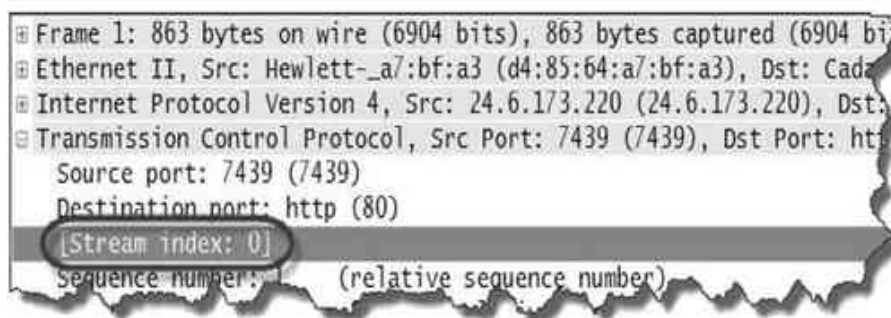
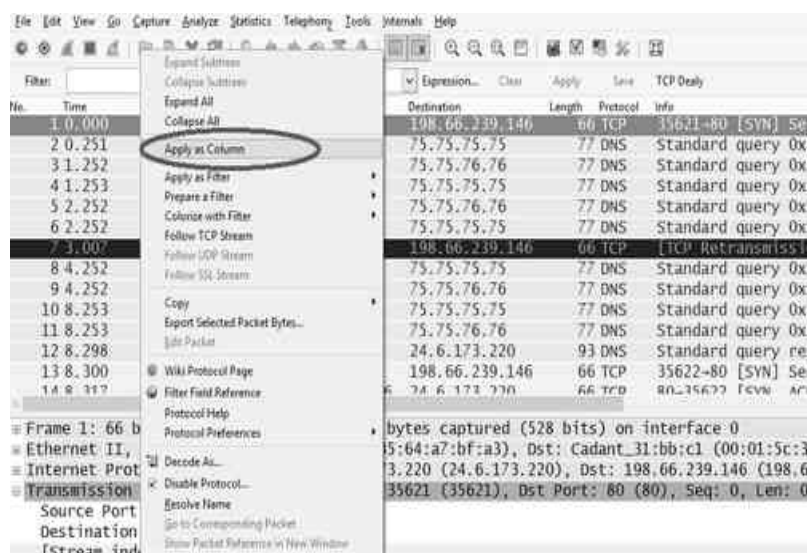


Figure 120.1: Add a TCP Stream Index Column

Step 1: Open *tr-chappellu.pcapng*.

Step 2: Expand the TCP header in Packet 1. Right-click on the **[Stream index: 0]** line and select **Apply as Column**.



Step 3: Click on your **Stream index** column once to sort the trace file by conversations. Jump to the end of the trace file and you find that there are 23 TCP conversations. Counting TCP streams starts at 0.

n-chappell@pcping [Wireshark 1.12.3 [v1.12.3-0-gbb3e9a0 from master-1.12]]									
File Edit View Go Capture Analysis Statistics Telephony Tools Internals Help									
Filter:				Expression...		Clear Apply Save		TCP-HS	
No.	Time	Delta	Delta Displayed	Source	Destination	Length	Protocol	Stream index	Info
693	34.0840.00017	0.000		24.6.173.220	207.171.187	54	TCP	22 35644-80	[ACK] Seq=...
692	34.0840.00000	0.000		207.171.187	24.6.173.22	359	TCP	22 80-35644	[PSH, ACK]
691	34.0840.00582	0.005		207.171.187	24.6.173.22	1514	TCP	22 80-35644	[ACK] Seq=...
690	34.0780.00013	0.000		24.6.173.220	207.171.187	54	TCP	22 35644-80	[ACK] Seq=...
689	34.0780.00000	0.000		207.171.187	24.6.173.22	1514	TCP	22 80-35644	[ACK] Seq=...
688	34.0780.00089	0.000		207.171.187	24.6.173.22	1514	TCP	22 80-35644	[ACK] Seq=...
687	34.0770.00271	0.002		207.171.187	24.6.173.22	1514	TCP	22 80-35644	[ACK] Seq=...
686	34.0750.00018	0.000		24.6.173.220	207.171.187	54	TCP	22 35644-80	[ACK] Seq=...
685	34.0740.00000	0.000		207.171.187	24.6.173.22	1514	TCP	22 80-35644	[ACK] Seq=...
684	34.0740.00082	0.000		207.171.187	24.6.173.22	1514	TCP	22 80-35644	[ACK] Seq=...
683	34.0720.00010	0.000		24.6.173.220	207.171.187	54	TCP	22 35644-80	[ACK] Seq=...

Figures and Material used for Part2 (from Topics 121 to 172) have been adapted from “Troubleshooting with Wireshark: Locate the Source of Performance Problems”, 2014, by L. Chappell.

Topic 121: Wireshark Lab 29

This topic describes how to add/sort a TCP Delta Time column in Wireshark.

The **tcp.time_delta** shows the time from the end of one packet in a TCP stream to the end of the next packet in that same TCP stream. To add the **tcp.time_delta** column, you must enable the TCP **Calculate conversation timestamp** preference. Click the **Preferences** button, expand **Protocols** and choose **TCP**.

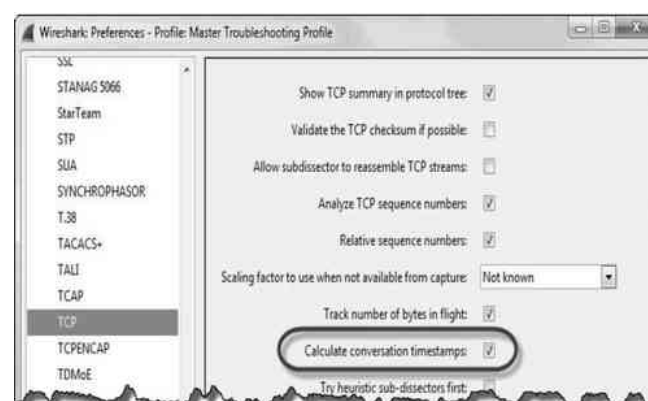


Figure 121.1: Add/Sort a TCP Delta Time Column

Step 1: Open *tr-chappellu.pcapng*.

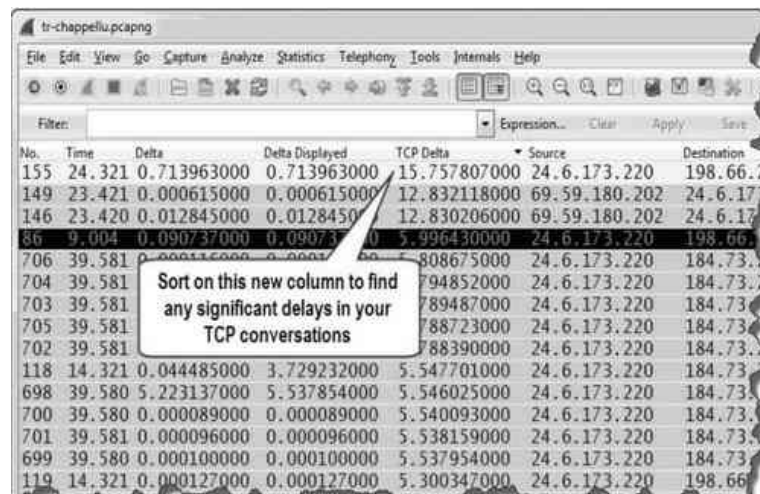
Step 2: Expand the **TCP header** in Packet 1. Right click anywhere on the TCP header, select **Protocol Preferences** and ensure that *Calculate conversation timestamps* is enabled.

Step 3: At the end of the TCP header, go to the **[Timestamps]** section, locate and right-click on the **Time since previous frame in this TCP stream** field. Select **Apply as Column**.

Step 4: The newly created column appears to the left of the Info column. Click and drag your new column to the right of the existing **Delta Displayed** column.

Step 5: Update the column name to **TCP Delta** by right-clicking on the column heading and selecting **Edit Column Details**. Click **OK**.

Step 6: Click the new **TCP Delta** column heading twice to sort from high to low. The packets with the largest delays before them in a TCP conversation appear at the top of the list.



No.	Time	Delta	Delta Displayed	TCP Delta	Source	Destination
155	24.321	0.713963000	0.713963000	15.757807000	24.6.173.220	198.66.2
149	23.421	0.000615000	0.000615000	12.832118000	69.59.180.202	24.6.173
146	23.420	0.012845000	0.012845000	12.830206000	69.59.180.202	24.6.173
86	9.004	0.090737000	0.090737000	5.996430000	24.6.173.220	198.66.2
706	39.581	0.000000000	0.000000000	808675000	24.6.173.220	184.73.2
704	39.581	0.000000000	0.000000000	794852000	24.6.173.220	184.73.2
703	39.581	0.000000000	0.000000000	789487000	24.6.173.220	184.73.2
705	39.581	0.000000000	0.000000000	788723000	24.6.173.220	184.73.2
702	39.581	0.000000000	0.000000000	788390000	24.6.173.220	184.73.2
118	14.321	0.044485000	3.729232000	5.547701000	24.6.173.220	184.73.2
698	39.580	5.223137000	5.537854000	5.546025000	24.6.173.220	184.73.2
700	39.580	0.000089000	0.000089000	5.540093000	24.6.173.220	184.73.2
701	39.581	0.000096000	0.000096000	5.538159000	24.6.173.220	184.73.2
699	39.580	0.000100000	0.000100000	5.537954000	24.6.173.220	184.73.2
119	14.321	0.000127000	0.000127000	5.300347000	24.6.173.220	198.66.2

Step 7: Scroll through the packets to determine where delays have incurred in this communication.

Topic 122: Wireshark Lab 30

This topic describes how to add a TCP Delay Button in Wireshark.

Using a "TCP Delay" button (filter expression) will allow us to efficiently locate delays. By pressing button TCP packets that are preceded by a noticeable delay will be displayed.

Step 1: Open *tr-chappellu.pcapng*.

Step 2: In the display filter area, enter the following filter: **tcp.time_delta > 1**

Step 3: Click the **Save** button on the display filter toolbar. Enter **TCP Delay** as the label when prompted. Click **OK** to save your new button.

Step 4: Click your new **TCP Delay** button. You will find that 37 packets match the filter. There are numerous TCP FIN packets in the list – we do not care about it.

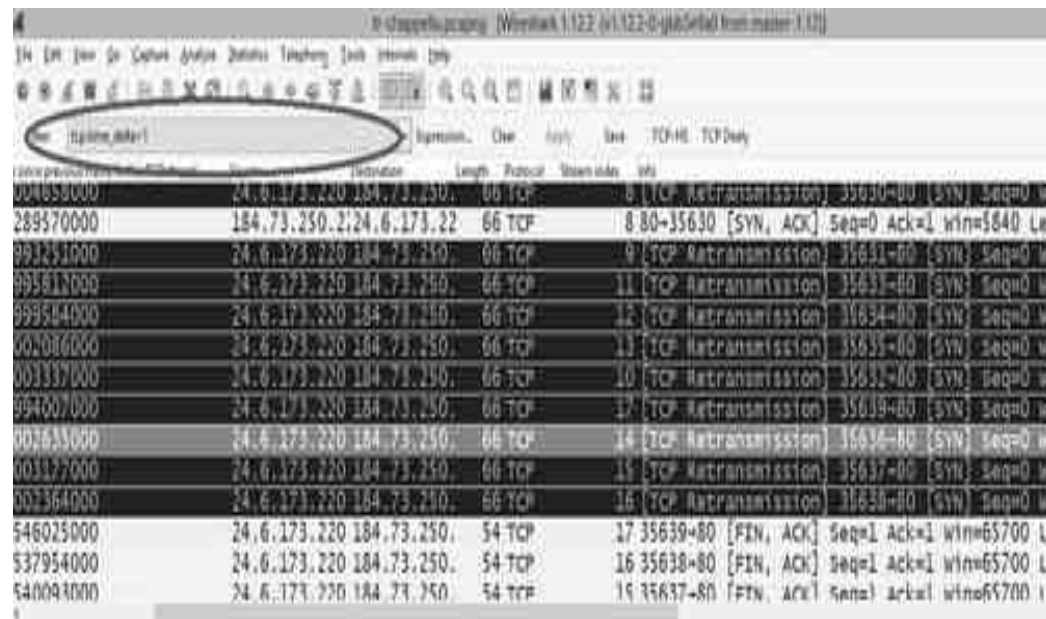
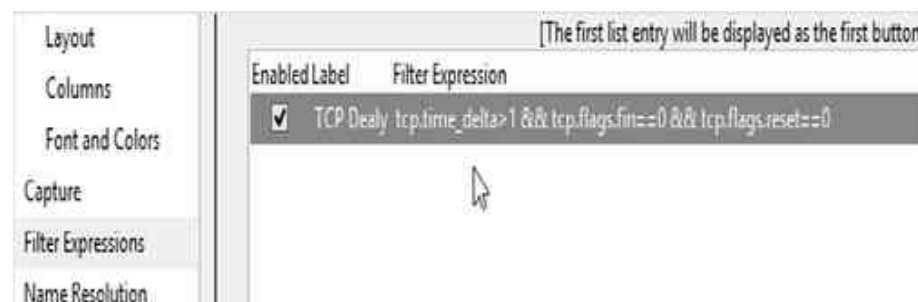


Figure 122.1: Add a "TCP Delay" Button

Step 5: Select **Edit | Preferences | Filter Expressions**, update TCP Delay filter expression to: **tcp.time_delta > 1 && tcp.flags.fin==0 && tcp.flags.reset==0** and then Click **OK**.



Step 6: Click your TCP Delay button again. 23 packets are displayed because TCP FIN and RST packets have been removed. Let's further remove HTTP GET requests from the TCP Delay button. Add the following string to the end of your filter: **&& !http.request.method=="GET"**. The highest TCP Delta delay is under 6 seconds and is a SYN retransmission pkt. There are 12 SYN retransmissions between the client and 184.73.250.227. There is one SYN/ACK as the RTT is 1.28957 seconds.

No.	Time	TCP Delta	Source	Destination	Length	Protocol	Stream index	Info
86	9.004	3.996430000	24.6.173.22	198.66.239	62	TCP	0	[TCP Retransmission]
7	3.007	3.007726000	24.6.173.22	198.66.239	66	TCP	0	[TCP Retransmission]
189	31.41	3.004658000	24.6.173.22	184.73.250	66	TCP	8	[TCP Retransmission]
554	33.69	3.003337000	24.6.173.22	184.73.250	66	TCP	10	[TCP Retransmission]
637	33.94	3.003177000	24.6.173.22	184.73.250	66	TCP	15	[TCP Retransmission]
636	33.94	3.002635000	24.6.173.22	184.73.250	66	TCP	14	[TCP Retransmission]
638	33.94	3.002364000	24.6.173.22	184.73.250	66	TCP	16	[TCP Retransmission]
553	33.69	3.002086000	24.6.173.22	184.73.250	66	TCP	13	[TCP Retransmission]
551	33.69	2.999584000	24.6.173.22	184.73.250	66	TCP	12	[TCP Retransmission]
550	33.68	2.995812000	24.6.173.22	184.73.250	66	TCP	11	[TCP Retransmission]
635	33.93	2.994007000	24.6.173.22	184.73.250	66	TCP	17	[TCP Retransmission]
535	33.67	2.993251000	24.6.173.22	184.73.250	66	TCP	9	[TCP Retransmission]

Topic 123: Wireshark Lab 31

This topic discusses how to obtain the round trip time (RTT) using TCP handshake in Wireshark.

A file transfer process will be very slow if RTT is extremely high and everything else is functioning properly. Wireshark allows us to estimate the RTT by looking at the **tcp.time_delta** value. We can capture at client side and look at the **tcp.time_delta** value between the client's TCP SYN packet and the server's TCP SYN/ACK response.

Step 1: Open *tr-cnn.pcapng*

Step 2: Enter **tcp.flags.syn==1** in the display filter area and then click **Apply**.

Step 3: Click your TCP Delta column heading twice to sort from high to low. We are interested in the delays preceding SYN/ACK packets.